Applications of Game Theory in Information Security

Viet Hoang Pham

Thesis submitted to
Royal Holloway, University of London
for the degree of
Doctor of Philosophy

2015

Declaration

All results presented in this thesis were produced under the supervision of Prof. Carlos Cid, during the course of study toward a PhD Information Security at Royal Holloway, University of London. They have not been submitted for any other degree in any other university or education establishment.

All research are carried out by myself and partly in collaboration with others. Some of the results were previously published in the following publications, in which all authors contributed equally:

- Viet Pham and Carlos Cid. Are we compromised? Modelling security assessment games. In *Decision and Game Theory for Security*, pages 234-247. Springer, 2012.
- MHR Khouzani, Viet Pham and Carlos Cid. Incentive engineering for outsourced computation in the face of collusion. *Workshop on the Economics of Information Security*. 2014.
- Viet Pham, MHR Khouzani and Carlos Cid. Optimal contracts for outsourced computation. *Decision and Game Theory for Security*. 2014.
- MHR Khouzani, Viet Pham and Carlos Cid. Strategic discovery and sharing of vulnerabilities in competitive environments. *Decision and Game Theory for Security*. 2014.

Other than these, the remaining part of the thesis comes from my own original research.

Signed: .....
(Viet Hoang Pham)

Date: **09 May 2015**

Acknowledgement

I would like to thank my supervisor, Professor Carlos Cid, for his excellent support and guidance throughout my studies at Royal Holloway. He has enlightened me with many insights an expert with complete view of the field would have acquired. Beside supervision, he is also a kind friend who is always available for helps and advices. I am deeply indebted to him for all his contributions to the completion of this thesis.

My collaborator and friend, Dr. Arman Khouzani, has also been of great helps during my research and studies. His experience and knowledge have enabled our successes in pursuing many exciting results.

Foremost, I am thankful to my parents for their unlimited love, and support throughout my whole life. Their continuous encouragement and faith have always been the main sources of my courage and inspiration.

My fiancee, Penying Rochanakul, has always been side-by-side to share both my joys and challenges. I deeply appreciate her selfless time, care and love as essential and never-fading supplies for my energetic life.

I am grateful to all my officemates in McCrea 355 since the past four years and now, as well as others in the department of Mathematics. We have together created an excellent environment, both academically and socially.

Finally, I would like to thank the college for its financial sponsorship, without which I would not be able to commence and carry on this PhD study.

Abstract

A new trend of research in information security revolves around the idea of treating individuals not as their intrinsic characteristics, e.g., honest or dishonest, but as utility maximisers. This is a special feature of the field of economics of security, namely rational security. Looking into the economic incentives of participants in a security scenario brings different insights and solutions than traditional security research in cryptography or formal method. First, traditional security mechanisms assume a set of permanently honest parties, which does not necessarily hold in economic models with utility-driven behaviours. Second, the notion of capabilities/powers/advantages of dishonest parties in traditional mechanisms may be too strong for certain scenarios (e.g., many civil purposes), leading to either impossibility results or practically infeasible security solutions.

In this thesis, we examine several security problems where above issues would emerge alongside traditional security research. We use game theory to study strategies and economic incentives of participants in these problems, e.g., attackers and defenders. Our goal is to provide, for each scenario, useful insights about the trend of behaviours/decisions these participants should take, which would be useful in understanding and predicting their actual courses of actions, thus helping future research or realistic solution design. When possible, we also propose security solutions, such as protocols or contracts that, under rational security, would lead to desirable outcomes in which, for example, attacks do not occur. Our research involves both high-level (e.g., investment) and low-level (e.g., network communication) security problems.

Contents

List of Figures

List of Tables

List of Notations

$\Delta(\cdot)$, $\mathcal{D}(\cdot)$	probability distributions over set
BR_i	best-response function of player i
$x\|\|y$	the concatenation of x and y
$\|M\|$	length of bit string M
$\{0,1\}^n$	all binary strings of length n
$\{0,1\}^*$	all binary strings
$[n]$	the set of natural numbers $\{1,\ldots,n\}$
$\mathbb{E}[\cdot]$	expected value
$\boldsymbol{o} \leftarrow \boldsymbol{A}$	output(s) $\boldsymbol{o}$ of PPT algorithm(s) $\boldsymbol{A}$
$e \leftarrow_{\$} S$	uniform sampling of a set S for element $e \in S$
$e \leftarrow_{\mathcal{D}} S$	sampling of a set S for element $e \in S$ with distribution $\mathcal{D}$
$\Pr_{\mathcal{D}}[e]$	probability of element e within probability space $\mathcal{D}$
$\Pr[\mathsf{e} : \mathsf{o} \mid \mathsf{c}]$	probability that an experiment e gives outcome o given condition c

Chapter 1

Introduction

The concept of rationality-driven behaviours has long been subliminally acknowledged and accepted in human society. As the main tool for modelling and analysing such behaviours within interactive environments, game-theoretic reasoning has brought significant contributions to the fundamentals of economics. It also pervades to other fields such as political sciences, biology, and is potentially applicable to all problem contexts with multiple decision-makers of unaligned interests. In a typical example of such, game-theoretic studies of information security problems has recently emerged as a new research direction, spanning across many subfields such as cryptography, network security, privacy, etc. In this chapter we explain the advantage of this direction and therefore motivate our interests in applications of game theory to information security. For the purpose of self-containment, we also discuss basic game concepts that will occur frequently throughout the content of the thesis.

1.1 Motivation

Our research is motivated by the fact that traditional treatments of security problems may be too strict to adapt to the dynamic changes of reality, which are often led by economic incentives. We explain this further by contrasting the two different views on participants in a security problem, namely *honest/malicious* versus *rational* agents.

1.1.1 The Nature of Information Security

The world nowadays values information as one of the most important and strategic assets for the survival and growth of communities and businesses [130]. As information technology evolves, a wide range of systems are being utilised, from personal devices such as PCs and mobile phones, to sophisticated server farms, mainframes, or super-

computers (for organisations). The better collection, refinement and effective processing make information even more valuable as a resource. Hence, whilst businesses gain remarkable benefit through effective gathering and use of information, the lack of such effectiveness would also lead to disruptions, delays, or failures in operations.

The need to protect such effectiveness from deliberate corruptions gives rise to what is called *information security*, termed in US Code, title 44, section 3542 as "protecting information and information systems from unauthorised access, use, disclosure, disruption, modification, or destruction..." [140]. There are two important observations from this definition, namely the act of protecting, and what to protect from. The need for information security thus implies the existence of threats, such as malicious attackers, to information and/or information systems. The existence of threats in turn leads to the emergence of defenders in response to the need of protection.

From the above description, information security conveys a set of games, or interactive situations, between the attackers (who might be referred to as "bad guys" based on social knowledge) and the defenders, the "good guys". Alternatively these two sides may be termed malicious and honest agents, respectively. Depending on the specific types of targets, attackers can also be of different types. For example, there are organisational insiders who seek sensitive/secret information over illegal accesses to database or wiretapping communication channels, spies that aims at sabotaging information infrastructures, or simply script-kiddies digging around the Internet for trivially vulnerable websites, etc. For each type of attacker, a corresponding defender exists, perhaps not restricted to prevent attacks from happening, but to minimise the risk-damage product caused by such attacks.

A traditional view of security problems is that for each scenario there is a clearly defined set of attackers and defenders whose types and purposes remain unchanged over time. To account for the worst case, security solutions are often designed so that even the most powerful adversaries, e.g., ones with infinite power, would gain little by committing the attack. Such solutions are then utilised in all scenarios, thanks to the security guarantees they hold. Notable examples of such attitude are the seminal model of mutually authenticated communication by Bellare and Rogaway [17] and the Dolev-Yao threat model [47]. Another distinguished example is the general concept of multiparty protocols, such as multiparty computation [95] and secret sharing [135], in which parties are either honest or dishonest. Although the context does not explicitly define the nature of each party, any existing security solution operates on a specific configuration of their honesty.

1.1.2 The Role of Rationality

Whether provably secure, traditional treatment of security might be problematic when bridging from theory into practice. In fact, there might be cases in which strongly secure solutions are either too expensive or restrictive for deployments, especially if the threat level is low, i.e., the solution is an overkill. For example, the *Public Key Infrastructure* (PKI), although has been cryptographically implemented and is thus strongly secure, is not universally deployed for securing Internet communications. This could be due to the unaffordable cost of adopting PKI, e.g., in peer-to-peer network of Internet users. Alternatively, the communication (e.g. news surfing) may be not sensitive enough to require such extreme guarantee of security.

Even if technical security solutions are being employed, the overall security also depends on the decision making process of users of such solutions. This becomes more and more significant as the involvement of users' decision increase, for example in *leap-of-faith* protocols [112]. Here a user must decide whether to trust the peer it is talking with. The user may be careful enough to attempt to verify the peer's trustworthiness, or it might just be lazy and forgo the checking. Another notorious example is phone-to-phone communication via Bluetooth, in which phone holders are supposed to match the random strings appeared on each other's screen [146]. Lazy users might skip this step and are thus exposed to the threat of *man-in-the-middle* attacks [33].

Another problem not captured in traditional security solutions is the dynamic changes in the types of agents, e.g., from honest to dishonest/malicious. This can be easily seen in multiparty protocols that assume the honesty of certain parties. In reality, any party might as well deviate from its supposed honesty if there is enough benefit in doing so. For example, Halpern and Teague [67] argue that a party in multiparty computation may withdraw its participation as soon as it receives the computation result, thus leaving the computation process incomplete. Since no party can be fully trusted to be honest, existing solutions would fail to deliver their security guarantees. A similar issue also occurs in attacker-defender paradigm. An attacker may cease its attack if that is costly whilst the gain is small. Meanwhile, misaligned incentives [6] may neutralise security measures because defenders are not motivated enough to utilise them properly.

In one way, we may argue that agents' behaviours in traditional security treatments are *irrational*, that is, they act in a way against their preferences. For example, the adversary model for a cryptographic protocol used in daily life communication (such as TLS) is assumed to be very powerful, e.g., it is willing to spend its effort breaking a 1024-bit RSA key. However, except for critical-mission purposes, the cost of breaking this key may not be paid well, if such key is only used for a small HTTP server.

Likewise, the assumed honest/dishonest parties in multiparty protocols, as well as those of misaligned incentives [6] are all seemingly irrational. From economic viewpoint however, this phenomenon is instead termed *bounded rationality* [57]. In other words, agents are in fact still rational, but their knowledge and capacity is bounded that lead to inaccurate analysis of the situations, and hence suboptimal decision making.

To address this issue, one needs to revise the concept of *rationality* and re-model agents following this concept [5]. It is not difficult to realise that in not just security, a rational agent must have preferences over its choices. Such preferences could, for example be economic benefits, morality, reputation, etc, as consequences of the chosen behaviour. Based on a well-defined system of preferences, a rational agent would behave in the most preferred way. In contexts like security scenarios however, the situation might be further complicated, because an agent's outcome (which determine preferences) depends not just on its behaviour, but also other agents involved (e.g., attackers, collaborators). The rationality of an agent must also take into account these externalities.

Note that in the description above, we do not mention security as part of rationality. Thus, when adopting rationality, the "security battle" among honest and malicious agents is more like a game where security is really not the matter, but that each game player selects an action/strategy/move[1] in a way aligned with its preferences, and that such preferences persist over time. By modelling agents as being rational instead of having definite type (e.g., honest, dishonest), we would thus be able to capture their actual behaviours in each realistic scenario where the security game is played. Consequently, this leads to our interest in using *game theory*, a rich mathematical toolset for modelling rational agents, to model and analyse security problems, as well as to design rational solutions.

1.2 Basics Concepts in Game Theory

Subliminal existence of game theory appeared as early as the 18th century [19]. It was, however, not widely studied and developed as a field of mathematics, until early 20th century, with notable work by von Neumann and Morgenstern [103]. Game theoretic studies focus on situations involving more than one party, with unaligned interests. In other words, a party's outcome is influenced not only by his/her actions, but also by the choices of others. As a result, such situations are referred to as *games*.

Game-theoretic studies involve formalisation of the context in which the game is

[1] These are used with the same meaning, in different texts: what a player chooses to do in a strategic-form game. For consistency we denote by action what a player can actually perform, and by strategy how the action is chosen, e.g., a probability distribution over an action space.

played, as well as analysis of players' strategic behaviours driven by their rationality. Formal models often consist of several important components, including a list of players, their available actions/strategies, their preferences over outcomes resulted from chosen strategies. For convenience of analysis we assume that players' preferences can be effectively represented by assigning real values to outcomes. Such assignment is called a *utility function*, and is generally unique for each player. Results from analysis of game models could be used in various ways, e.g., as recommendations for players, as a guide for altering the context so as to drive the strategic behaviours of the players. In the game-theoretic jargon, the latter is referred to as *mechanism design*.

In the following subsections we discuss several important game concepts which are used in our research. These are strategic-form games with players' simultaneous moves, extensive-form games with sequential moves [83], and incomplete information games [68] in which players might be uncertain about the game structure, such as payoff functions, or others' set of strategies. For each game we also discuss important equilibrium concepts that are useful for the purpose of game analysis and mechanism designs. Note that for simplicity of presentation we only consider finite games, i.e., games with finite strategy sets/spaces. The definition for games with sets of infinite strategies can be adapted easily as needed.

1.2.1 Strategic-Form Games

The simplest but also central game concept is *one-shot*, or *strategic form*, or simultaneous games. Often these games model situations in which players move simultaneously. However, in reality there is hardly any situation in which moves are made absolutely simultaneously. A better description of these games is that players play independently. In other words, each player obverses no information about others' decisions, and hence independently selects a *pure strategy*[2]. Thus, whether moves are simultaneous of sequential does not matter. The game can be defined formally as follows:

Definition 1.1. *A strategic-form game is a tuple* $\langle N, \{A_i\}_{i\in N}, \{u_i\}_{i\in N}\rangle$ *where* N *is the set of players, for each* $i \in N$, A_i *is a non-empty non-singleton set of pure strategies available for player* i, *and* $u_i : A \to \mathbb{R}$ *is player* i*'s utility/payoff function, where* $A = \times_{i\in N} A_i$. *For each strategy profile* $\boldsymbol{a} = (\boldsymbol{a}_i)_{i\in N} \in A$ *and* $i \in N$, *denote* $u_i(\boldsymbol{a}_i, \boldsymbol{a}_{-i}) = u_i(\boldsymbol{a})$, *where* $\boldsymbol{a}_{-i} = (\boldsymbol{a}_j)_{j\neq i}$.

Although simple, the above definition captures all the necessary information of a game. Indeed, each tuple $\boldsymbol{a} = (\boldsymbol{a}_i)_{i\in N} \in A$, which can be called as either *strategy*

[2]A pure strategy is a deterministic selection of actions, i.e., it is equivalent to an action. This is in contrast with a *mixed* strategy that chooses an action probabilistically.

profile or *outcome*, represents what may happen at the end, whereas A is the set of all possible outcomes. The rationality of each player is depicted via its utility function, so that for two outcomes $\boldsymbol{a}, \boldsymbol{a}' \in A$ with $u_i(\boldsymbol{a}) > u_i(\boldsymbol{a}')$, it means that player i *prefers* $\boldsymbol{a}$ to $\boldsymbol{a}'$. Inherently, each player must know exactly about what others can do ($\{A_i\}$) as well as their preferences ($\{u_i\}$).

To support analysis of games, a number of *solution concepts* have been constructed. Basically, solution concepts refer to strategies/outcomes with some particular feature(s) which can be either interesting, or desirable. A fundamental notion in non-cooperative game theory is the *best-response*:

Definition 1.2. *Let $\langle N, \{A_i\}_{i\in N}, \{u_i\}_{i\in N}\rangle$ be a strategic-form game. For each player $i \in N$ let $A_{-i} = \times_{j\neq i} A_j$, player i's best-response $\mathsf{BR}_i : A_{-i} \to 2^{A_i}$ is defined as*

$$\mathsf{BR}_i(\boldsymbol{a}_{-i}) = \arg\max_{a_i \in A_i} u_i(a_i, \boldsymbol{a}_{-i})$$

While the best-response function simply states which action(s) a player should follow in response to specific action tuples by the rest, it leads to more interesting solution concepts, e.g., *dominant equilibrium* and *Nash equilibrium*. Dominant equilibrium is an outcome in which no player needs to care about how others may act, as its chosen action is always the best response. This is probably the most desirable feature of any game, because if a game with this feature happens in real life, such outcome is very likely to occur. Nash equilibrium, a central breakthrough in game theory however, relaxes the former "difficult-to-achieve" solution, in that each player's action needs to be a best-response to only what others have chosen in the outcome. In other words, if the outcome has already been designated to occur, then no individual player would benefit by acting in a different way. These two concepts are formalised as follows:

Definition 1.3. *Let $\langle N, \{A_i\}_{i\in N}, \{u_i\}_{i\in N}\rangle$ be a strategic-form game. For each player $i \in N$, a strategy $a_i \in A_i$ is dominant if $a_i \in \cap_{\boldsymbol{a}_{-i}\in A_{-i}} \mathsf{BR}_i(\boldsymbol{a}_{-i})$. An outcome $\boldsymbol{a} \in A$ is a dominant equilibrium if for each $i \in N$, $a_i \in \boldsymbol{a}$ is a dominant strategy of player i.*

Definition 1.4. *Let $\langle N, \{A_i\}_{i\in N}, \{u_i\}_{i\in N}\rangle$ be a strategic-form game. An outcome $\boldsymbol{a} \in A$ is a Nash equilibrium if $a_i \in \mathsf{BR}_i(\boldsymbol{a}_{-i})$ for all $i \in N$.*

The reason why some outcome is called an equilibrium is because each player, knowing exactly the actions that will be taken by others, do not want to deviate from the "equilibrium". This left out other interesting equilibria in which such knowledge is imperfect, i.e., each player only has knowledge on the probability distribution over the actions of others. This leads to the concept of *mixed-strategy* version of a game, made famous by von Neumann and Morgenstern [103]:

Definition 1.5. *Let* $\Gamma = \langle N, \{A_i\}_{i\in N}, \{u_i\}_{i\in N}\rangle$ *be a strategic-form game such that* A *is finite. A mixed-strategy version of* Γ *is a strategic-form game* $\langle N, \{\mathcal{A}_i\}_{i\in N}, \{U_i\}_{i\in N}\rangle$ *such that for each player* $i \in N$, $\mathcal{A}_i$ *is the set of all probability distribution over* A_i, *and that for all* $(\Delta_i(A_i))_{i\in N} \in \times_{i\in N}\mathcal{A}_i = \mathcal{A}$,

$$U_i((\Delta_i(A_i))_{i\in N}) = \sum_{\boldsymbol{a}\in A}\left(\prod_{i\in N}\Pr_{\Delta_i}[\boldsymbol{a}_i]\right)u_i(\boldsymbol{a})$$

An interesting result with mixed-strategy version of a game with finite actions is that there always exists a Nash equilibrium [103]. When each player' actions form a compact metric space, Glicksberg [59] extends Kakutani's fixed point theorem [76] and shows the same conclusion, i.e., there is always a Nash equilibrium. Note here, that the mixed-strategy version of a mixed-strategy game is of no difference, since eventually each player's strategy is a probability measure over the set of deterministic actions.

1.2.2 Incomplete Information Bayesian Games

Incomplete information refers to situations in which there is an oblivion by at least one player about the exact outcome preferences of either itself or some other. This happens due to the existence of the so-called *state of the Nature*, which is decided probabilistically by some relevant random source, which we call the *Nature*. For example, consider a game played among a large population, in which each player must decide whether to buy a lottery ticket. Even if a player knows every other's decision, it cannot determine its own utility, which is the earning from the lottery subtracted by the ticket price. This is because the outcome of the lottery is probabilistically decided by the *Nature*, which in this case is the lottery company. In security games, although it is known that every player is rational, an "evil mind" player would have different preferences to a "benign" player. Such *type* of a player is often unknown to the others, and is decided probabilistically by Nature whose distribution depends on, for example, living environments, education background, etc.

The simplest appearance of the Nature is in Bayesian games, which is a strategic-form game, except that the utility function of each player probabilistically depends on the state of the Nature. Note here that the probability distribution over these states is private to each player i (depicted by p_i), which represents its knowledge about the Nature. To be even more generic, Bayesian games allow the Nature to leak some information about its state, called *signal*, to players, so that they can probabilistically work out their expected utilities. This comes from the fact that in reality, even if no signal is given, data acquisition and statistical analysis might be used to learn about

the state of the Nature. Bayesian games are defined in the following:

Definition 1.6. *A Bayesian game is a tuple* $\langle N, \Omega, \langle A_i, T_i, C_i, \tau_i, p_i, u_i \rangle_{i \in N} \rangle$ *where* Ω *is the set of nature states, and for each* $i \in N$*,*

- A_i *is the set of* i*'s all available actions,*
- T_i *is the set of* i*'s signals/types, with* $\tau_i : \Omega \to T_i$ *is the state-to-signal mapping,*
- $C_i : T_i \to 2^{A_i}$ *is the set of* i*'s available actions after receiving* $t_i \in T_i$*,*
- p_i *is a probability measure over* Ω*, and,*
- $u_i : \Omega \times A \to \mathbb{R}$ *is player* i*'s utility function*

The adaptation of Nash equilibrium into Bayesian games, called *Bayesian Nash equilibrium*, naturally follows from Definition 1.5:

Definition 1.7. *Let* $G = \langle N, \Omega, \langle A_i, T_i, C_i, \tau_i, p_i, u_i \rangle_{i \in N} \rangle$ *be a Bayesian game. A strategic form of* G *is a tuple* $\langle N, \{\mathcal{A}_i\}_{i \in N}, \{U_i\}_{i \in N} \rangle$ *where for each* $i \in N$*,* $\mathcal{A}_i = \times_{t \in T_i} C_i(t)$*, and denote* $a_i \in \mathcal{A}_i$ *of the form* $a_i : T_i \to A_i$*, and*

$$U_i(\boldsymbol{a}) = \sum_{\omega \in \Omega} p_i(\omega) u_i(\omega, (\boldsymbol{a}_j(\tau_j(\omega)))_{i \in N})$$

for all $\boldsymbol{a} \in \times_{i \in N} \mathcal{A}_i$*.*

Definition 1.8. *Let* $G = \langle N, \Omega, \langle A_i, T_i, C_i, \tau_i, p_i, u_i \rangle_{i \in N} \rangle$ *be a Bayesian game and* $G' = \langle N, \{\mathcal{A}_i\}_{i \in N}, \{U_i\}_{i \in N} \rangle$ *be its strategic form. A Bayesian Nash equilibrium of* G *is a Nash equilibrium of the mixed-strategy version of* G'*.*

1.2.3 Extensive-Form Games

A fundamental assumption in strategic-form games is that players' actions are independent. This does not hold in many situations, e.g., in chess games where each player' moves are observable to its opponent. Such situations can alternatively be modelled in *extensive form*, a game tree that specifies the order of plays, as well as what each player can do at each of its turns. A general notion of extensive-form games also comes with *incomplete* (with the Nature participating in the game) and *imperfect* information, as defined below:

Definition 1.9. *An extensive-form game* Γ *is a tuple* $\langle N, \mathcal{K}, \boldsymbol{H}, \mathcal{A}, \rho, \{u_i\}_{i \in N} \rangle$ *where*

- N *is the set of players,*

- $\mathcal{A}$ *is the set of all possible actions,*
- $\mathcal{K} = \langle V, E, r, p\rangle$ *is a rooted tree with nodes* V*, node-to-player mapping* $r : V \to \{0\} \cup N$*, edges* $E = \{(v, v') : v \text{ is the successor}\}$*, and edge-to-action mapping* $p : E \to \mathcal{A}$ *such that* $p(v, v') = p(v, v'')$ *iff* $v' = v''$ *for all* $v, v', v'' \in V$*,*
- $\boldsymbol{H}$ *is a partition of* V *such that all nodes within each* ***information set*** $H \in \boldsymbol{H}$ *(i) belong the same player, i.e., there exists* $i \in \{0\} \cup N$ *such that* $r(v) = i$ *for all* $v \in H$ *(denoted by* $r(H)$*) and, (ii) have the same set of available actions, i.e., for all* $v, v' \in H$*,* $\{p((v, \cdot) \in E)\} = \{p((v', \cdot) \in E)\}$ *(denoted by* $A(H)$*),*
- $\rho = \{\rho_H : A(H) \to [0, 1] \mid H \in \boldsymbol{H} \wedge r(v) = 0 \ \forall\ v \in H\}$ *is the family of probability distributions over the states of the Nature and,*
- $u_i : T \to \mathbb{R}$*, where* $T \in V$ *is the set of terminal nodes.*

In essence, each information set $H \in \boldsymbol{H}$ represents the imperfection in knowledge of a player about what has happened in the history, i.e., it is oblivious about which node in H it is currently in. Therefore, the third condition of Definition 1.9 requires that a player's action must be the same for every node in the same information set. Because of the dependence in the course of actions, we also need to be clear about what exactly is a player's strategy in this game, and the utility it receives from doing so. This can be best captured by representing the game in strategic form:

Definition 1.10. *Let* $\Gamma = \langle N, \mathcal{K}, \boldsymbol{H}, \mathcal{A}, \rho, \{u_i\}_{i\in N}\rangle$ *be an extensive-form game. The strategic form of* Γ *is a tuple* $\langle N, \{A_i\}_{i\in N}, \{U_i\}_{i\in N}\rangle$ *such that*

- *for each* $i \in N$*,* $A_i = \{a_i : \boldsymbol{H} \to \mathcal{A}\}$ *where each* a_i *maps an information set* $H \in \boldsymbol{H}$ *with* $r(H) = i$ *to an action in* $A(H) \subseteq \mathcal{A}$*.*
- *for each outcome* $\boldsymbol{a} \in \times_{i\in N} A_i$*, define* $\boldsymbol{o_a} : E \to [0, 1]$ *such that for* $(v, v') \in E$*, if* $r(v) = 0$ *then* $\boldsymbol{o}(v, v') = \rho_{H_0}(p(v, v'))$ *for* $H_0 \ni v$*, else* $\boldsymbol{o}(v, v') = 1$ *when* $a_{r(H)}(H) = p(v, v')$ *for* $H \ni v$*, and 0 otherwise, then*

$$U_i(\boldsymbol{a}) = \sum_{t\in T} u_i(t) \prod_{(v,v')\in \mathsf{path}(t)} \boldsymbol{o_a}(v, v') \tag{1.1}$$

where $\mathsf{path}(t)$ *is the set of edges from the root to* t*.*

As can be seen, because players' actions in an extensive-form game are not independent, a *pure strategy* of a player must be a plan of actions for every possible situation that could happen during the game. Such plan can be made in advance,

and is thus independent of others' plans. While the formalisation of Nash equilibrium for extensive-form game is trivial with the above definition, it is noticeable that such equilibrium might have a problem, well-known as *non-credible threats*.

Indeed, from (1.1) we see that $u_i(t)$ only enters the utility if every edge leading to t is chosen with non-zero probability. This indicates a possibility that some part of the plans of actions does not affect the utility, and thus can be arbitrarily set. Thus, a player may plan for that part irrationally, i.e., selecting non best-response action, while still guaranteeing Nash equilibrium. Even worse, sometimes such irrationality is the only way to make sure that the current outcome is in equilibrium. This is what referred to as non-credible threat: a threat of acting in a way that would otherwise not occur when triggered. Hence, a Nash equilibrium with such threats is hardly realistic. A refinement of this, called *subgame-perfect equilibrium* (SPE), safely eliminates non-credible threats:

Definition 1.11. *Let $\Gamma = \langle N, \mathcal{K}, \boldsymbol{H}, \mathcal{A}, \rho, \{u_i\}_{i\in N}\rangle$ be an extensive-form game. A subgame of Γ, defined over a node $v \in H \in \boldsymbol{H}$ where H is singleton, is a tuple $\langle N, \mathcal{K}', \boldsymbol{H}, \mathcal{A}, \rho, \{u_i\}_{i\in N}\rangle$ such that $\mathcal{K}'$ is a subtree of $\mathcal{K}$ containing v and all its successors. An outcome $\boldsymbol{a}$ of the strategic form of Γ is said to be a subgame-perfect equilibrium if, for every subgame Γ' of Γ, the induced outcome of $\boldsymbol{a}$ on Γ' is a Nash equilibrium of the strategic form of Γ'.*

According the above definition, a subgame-perfect equilibrium is only a proper refinement of Nash equilibrium if there is at least one singleton information set in the game tree. In other words, in each subgame at least one player must know exactly what has happened in the history, which include players' exact past actions and all realised states of the Nature. However, there might be games in which all information sets are non-singleton, and thus the only valid subgame of the game tree is itself, which reduces any SPE to a Nash equilibrium. This motivates further refinements, and consequently led to the concept of *Perfect Bayesian equilibrium* (PBE). Unlike an SPE that requires one player with perfect knowledge of history for each subgame, a PBE only requires a player with a *belief* about history which is consistent (following Bayes' rule) with the strategy profile forming the equilibrium. Since the notion SPE appears in many results of our research, we describe it formally below:

Definition 1.12. *Let $\Gamma = \langle N, \mathcal{K}, \boldsymbol{H}, \mathcal{A}, \rho, \{u_i\}_{i\in N}\rangle$ be an extensive-form game, and $\boldsymbol{a}$ be a strategy profile of the strategic form of Γ. For every information set $H \in \boldsymbol{H}$, a **belief** of player $i = r(v)$ for $v \in H$ is a probability distribution over H. A **belief system** is the collection of beliefs for all information sets $H \in \boldsymbol{H}$. A belief system is said to be **consistent** with $\boldsymbol{a}$ if for every $H \in \boldsymbol{H}$ that can be reached with positive*

probability, the belief over H is identical to the posterior distribution over H derived from $\boldsymbol{a}$ using Bayes' rule.

Definition 1.13. *Let $\Gamma = \langle N, \mathcal{K}, \boldsymbol{H}, \mathcal{A}, \rho, \{u_i\}_{i \in N} \rangle$ be an extensive-form game. A strategy profile $\boldsymbol{a}$ of the strategic form of Γ and a belief system μ are said to form a Perfect Bayesian equilibrium if (i) μ is consistent with $\boldsymbol{a}$ and, (ii) for every information set $H \in \boldsymbol{H}$ with $i = r(H) \neq 0$, then player i's utility is maximal given a belief over H derived from μ and other players' strategies in $\boldsymbol{a}$, that is,*

$$\boldsymbol{a}_i \in \arg\max_{a_i \in A_i} \sum_{v^* \in H} \Pr_{\mu}[v^*] \sum_{t \in T_{v^*}} u_i(t) \prod_{(v,v') \in \mathsf{path}(v^*,t)} \boldsymbol{o}_{\boldsymbol{a}'}(v, v')$$

where $\boldsymbol{a}' = (a_i, \boldsymbol{a}_{-i})$, T_{v^*} is the set of terminal nodes of $\mathcal{K}$ preceded by v^*, $\mathsf{path}(v^*, t)$ is the set of edges from v^* to t, and $\boldsymbol{o}_\cdot$ is as defined in Definition 1.10.

For compatibility with game-theoretic literature, with respect to a PBE we call the information sets that can be reached with positive probabilities as *on the equilibrium path*, and otherwise as *off the equilibrium path*. Unlike previous solution concepts, the specification of PBE requires not just a strategy profile, but also a belief system. This is because for information sets that are off the equilibrium path, there might be many valid beliefs, each of which would lead to a distinct belief system. This gives rooms to even further refinements of PBE along the development of game theory. In this thesis however, we only utilise the notion of PBE, and thus we may safely ignore the need to specify a belief system for each PBE, as the strategy profile alone would convey the most important parts of the corresponding belief system.

1.3 Basic Cryptographic Primitives

Although cryptography is not the main focus of our research, in some of our works presented in this thesis we rely on cryptographic techniques to make sure that our game models are realistic. To complete the formal treatments of our solutions, in this section we define standard cryptographic primitives that would be used in our solutions, particularly in Chapter 4 (Section 4.8) and Chapter 5 (Section 5.6). We start by recalling the central notion of negligible function, which is then followed by the definitions of key exchange, proof-of-works, authenticated encryption, and non-malleable commitments. Note that the definitions for authenticated encryption and non-malleable commitment are borrowed from standard cryptographic definitions, meaning that there are existing constructions satisfying them. However, our definitions of key exchange and proof-of-works have been adapted to our needs. For completeness of solutions in Appendices C

and D we show constructions that satisfy such adaptations.

Definition 1.14. *A negligible function is* $\epsilon : \mathbb{N} \to \mathbb{R}$ *such that for every positive polynomial* poly *there exists an integer* $n_0 > 0$ *such that for all* $n \geq n_0$ *we have* $|\epsilon(n)| \leq 1/\mathsf{poly}(n)$*. A non-negligible function is a function which is not negligible.*

Definition 1.15. *Let* $A_1, \ldots A_n$ *be probabilistic polynomial-time (PPT) algorithms. A parallel execution of* $A_1, \ldots, A_n$ *that respectively output* $o_1, \ldots, o_n$ *is denoted by* $(o_1, \ldots, o_n) \leftarrow (A_1, \ldots, A_n)$*.*

Definition 1.16 ([30])**.** *Let* $\mathcal{K}$ *be a key space. A secure key-exchange protocol is a tuple of PPT algorithms* (I, R) *satisfying for some negligible function* ϵ *the following properties:*

- *Correctness:* $\Pr[(k_1, k_2) \leftarrow (\mathsf{I}(n), \mathsf{R}(n)) : k_1 = k_2 \in \mathcal{K}] = 1$*,*
- *Synchronisation: if the adversary modifies the communication, then it is hard to make endpoints agree on the same key, i.e., for all PPT adversaries* Adv

$$\Pr\left[(k_1, \cdot, k_2) \leftarrow (\mathsf{I}(n), \mathsf{Adv}, \mathsf{R}(n)) : k_1 = k_2 \neq \perp \;\mid \mathsf{tr}_I \neq \mathsf{tr}_R\right] \leq \epsilon(n)$$

  where tr_I *and* tr_R *are transcripts of messages during the communication in the views of* $\mathsf{I}(n)$ *and* $\mathsf{R}(n)$*, respectively,*

- *Key indistinguishability: the adversary must either make the endpoints accepting different keys, or else it has no knowledge about the agreed key, i.e., denote by* $\mathcal{D}$ *the distribution over* $\mathcal{K}$ *of* k *given that* $(k, \cdot, k) \leftarrow (\mathsf{I}(n), \mathsf{R}(n))$*, for all stateful*[3] *PPT adversaries* Adv *define*

$$\Pi^{\mathsf{Adv}}_{\mathsf{IND\text{-}KE}} = \left[(k_0, \cdot, k_2) \leftarrow (\mathsf{I}(n), \mathsf{Adv}, \mathsf{R}(n)); k_1 \leftarrow_{\mathcal{D}} \mathcal{K}; b \leftarrow_{\$} \{0,1\}; b' \leftarrow \mathsf{Adv}(k_b)\right],$$

  then either of the following holds:

$$\Pr\left[\Pi^{\mathsf{Adv}}_{\mathsf{IND\text{-}KE}} : b = b' \mid k_0 = k_2 \neq \perp\right] \leq 1/2 + \epsilon(n)$$
$$\Pr\left[\Pi^{\mathsf{Adv}}_{\mathsf{IND\text{-}KE}} : k_0 = k_2 \neq \perp\right] \leq \epsilon(n)$$

Definition 1.17. *Denote by* $\mathsf{cost}(\mathsf{Alg})$ *a realisation of the computational cost when executing a probabilistic algorithm* Alg*. A proof-of-work mechanism is a tuple* $(\mathsf{Prove}, \mathsf{Verify})$

[3] By "stateful" we mean that the adversary remembers what it has done within the same experiment. This is to simplify the presentation of the definition by avoiding messages passing between stages of an experiment.

of PPT algorithms satisfying for some negligible function ϵ that for all cost values $c \in [c_{\min}, c_{\max}]$, all proof identifiers $id \in \mathsf{ID}$ for some ID space ID, the following properties are satisfied:

- *Correctness:*

$$\Pr\left[\begin{array}{c}(sig_1, sig_2) \leftarrow (\mathsf{Prove}_n(c, id), \mathsf{Verify}_n(c, id)) : \\ sig_1 = sig_2 = \mathsf{true} \wedge \mathsf{cost}(\mathsf{Prove}_n(c, id)) \geq c - \epsilon(n)\end{array}\right] = 1 \text{ and,}$$

- *Verifiability: even with helps from past or current communication, any save in computation cost would result in a reduce in convincing the verifier by at least the same proportion, i.e., for all PPT algorithms* $\mathsf{Adv} = (\mathsf{Adv1}, \mathsf{Adv2})$ *and all probabilities p:*

$$\Pr\left[\begin{array}{c}s \leftarrow \mathsf{Adv1}^{\mathsf{Prove}_n(\cdot,\cdot)}(n, c, id); (\cdot, sig) \leftarrow (\mathsf{Adv2}^{\mathsf{Prove}_n(\cdot,\cdot)}(n, c, id, s), \\ \mathsf{Verify}_n(c, id)) : \mathsf{cost}(\mathsf{Adv2}) \leq c \cdot p \wedge sig = \mathsf{true}\end{array}\right]^{4} \leq p + \epsilon(n)$$

  where $\mathsf{Adv2}$ *is not allowed to trigger* Prove *with id, and* $\mathsf{cost}(\mathsf{Adv2})$ *ignores the executions of* $\mathsf{Prove}(\cdot, \cdot)$*.*

Definition 1.18. *With respect to a message space $\mathcal{M}$, an authenticated encryption scheme is a tuple of PPT algorithms $(\mathcal{K}, \mathsf{Enc}, \mathsf{Dec})$ satisfying for some negligible function ϵ the following properties:*

- *Correctness: for all $m \in \mathcal{M}$ and all $\Pr[k \leftarrow \mathcal{K}(n) : \mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m] = 1$,*
- IND-CPA*: the content of ciphertexts convey no information about the plaintexts, i.e., for all stateful PPT adversaries* Adv

$$\Pr\left[\begin{array}{c}k \leftarrow \mathcal{K}(n); (m_0, m_1) \leftarrow \mathsf{Adv}^{\mathsf{Enc}_k(\cdot)}; b \leftarrow_{\$} \{0, 1\}; \\ b' \leftarrow \mathsf{Adv}^{\mathsf{Enc}_k(\cdot)}(\mathsf{Enc}_k(m_b)) : b = b'\end{array}\right] \leq 1/2 + \epsilon(n)$$

- IND-CTXT*: it is hard for an adversary to produce a ciphertext (other than those produced by the sender) that can be accepted by the receiver, i.e., for all stateful*

[4] Here $\mathsf{Adv1}^{\mathsf{Prove}(\cdot,\cdot)}$ represents the fact that the adversary may rely on knowledge from past executions of this proof-of-work mechanism, and $\mathsf{Adv2}^{\mathsf{Prove}(\cdot,\cdot)}$ indicates that the adversary might try to exploit computational effort from the client, but for a different proof with $id' \neq id$.

PPT adversaries Adv

$$\Pr[k \leftarrow \mathcal{K}(n); c \leftarrow \mathsf{Adv}^{\mathsf{Enc}_k(\cdot)} : \mathsf{Dec}_k(c) \neq \perp \wedge c \notin C] \leq \epsilon(n)$$

where C *is the set of ciphertexts output by* $\mathsf{Enc}_k(\cdot)$ *to* Adv.

Definition 1.19 ([43]). *With respect to a message space* $\mathcal{M}$*, a non-malleable commitment scheme under* common reference string *(CRS) model is a tuple of PPT algorithms* $(\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open})$ *satisfying for some negligible function* ϵ *the following properties:*

- *Correctness: for all* $m \in \mathcal{M}$

$$\Pr[\mathsf{CK} \leftarrow \mathsf{Setup}(n) : \mathsf{Open}_{\mathsf{CK}}(\mathsf{Commit}_{\mathsf{CK}}(m)) = m] = 1,$$

- *Hiding: the commit value conveys no information about the message being committed, i.e., for all stateful PPT adversaries* Adv

$$\Pr\left[\begin{array}{c}\mathsf{CK} \leftarrow \mathsf{Setup}(n); (m_0, m_1) \leftarrow \mathsf{Adv}(\mathsf{CK}); b \leftarrow_{\$} \{0,1\}; \\ (c,d) \leftarrow \mathsf{Commit}_{\mathsf{CK}}(m_b); b' \leftarrow \mathsf{Adv}(c) : b = b'\end{array}\right] \leq 1/2 + \epsilon(n),$$

- *Binding: a commit value is only valid for one message, i.e., for all PPT adversaries* Adv

$$\Pr\left[\begin{array}{c}\mathsf{CK} \leftarrow \mathsf{Setup}(n); (c,d,d') \leftarrow \mathsf{Adv}(\mathsf{CK}); m \leftarrow \mathsf{Open}_{\mathsf{CK}}(c,d); \\ m' \leftarrow \mathsf{Open}_{\mathsf{CK}}(c,d') : m \neq m' \wedge m, m' \neq \perp\end{array}\right] \leq \epsilon(n)$$

- *Non-malleability (w.r.t opening): given a commit value of a message, until receiving the corresponding opening value, it is hard to commit to any other related message, i.e., for every stateful PPT adversary* Adv *there exists a PPT adversary* Adv' *such that* $p_1 - p_2 \leq \epsilon(n)$ *for all efficiently sampleable distribution* $\mathcal{D}$ *over* $\mathcal{M}$*, and all polynomial-time computable relation* R*, where*

$$p_1 = \Pr\left[\begin{array}{c}\mathsf{CK} \leftarrow \mathsf{Setup}(n); m \leftarrow_{\mathcal{D}} \mathcal{M}; (c_1, d_1) \leftarrow \mathsf{Commit}_{\mathsf{CK}}(m); c_2 \leftarrow \mathsf{Adv}(\mathsf{CK}, c_1); \\ d_2 \leftarrow \mathsf{Adv}(\mathsf{CK}, c_1, c_2, d_1); m' \leftarrow \mathsf{Open}_{\mathsf{CK}}(c_2, d_2) : c_1 \neq c_2 \wedge R(m, m') = 1\end{array}\right]$$

$$p_2 = \Pr[m \leftarrow_{\mathcal{D}} \mathcal{M}; m' \leftarrow \mathsf{Adv}' : R(m, m') = 1]$$

1.4 Examples of Applications

The diversity of problem contexts in information security suggests a rich set of opportunities for game-theoretic studies. Indeed, research on game modelling of security has established a wide range of results on different problems and areas. A major game model used in these works is the *attacker-defender* game [3, 3.1], which is played between some attacker(s) and defender(s) in a strict competition, i.e., they have opposite preferences over outcomes. Alternatively, many research also consider *defenders-only* games, in which the defenders try to collaborate (whilst being individually selfish) in mitigating a particular source of threat. As an illustration, we briefly review some of the most notable game-theoretic studies on network security and security investment – the areas that draw the most attention from intellectual fore.

1.4.1 Network Security

Network security has witnessed enormous efforts in game-theoretic modelling of security. This is partly due to the diversity in problem contexts it covers, and party because these contexts often involve multiple participants with independent behaviours and conflict of interests, e.g., in client-server model, peer-to-peer model, and a wide range of adversary models. The most common approach to modelling network security problems is to use attacker-defender game, for example *zero-sum* games [121] as in [2,27,91,153], that is, the total utility of the players are always zero. Attacker-defender games can also be extended to multiple defenders or attackers [4,123].

Game-theoretic studies of security at TCP/IP access layer is one of the most active approaches. There are work that rely on the attacker-defender model, for example by Kashyap et al. [78], Altman et al. [4], or Buchegger and Alpan [27], who address the problem of signal jamming in wireless networks. Effective, efficient, and secure signal transmission is also investigated in defenders-only type of games, such as in the work of Sagduyu and Ephremides [128], Raya et al. [123]. Unlike other work that rely on non-cooperative players, Saad et al. [127] use *coalitional games* to study cooperation among defenders in minimising eavesdropping.

Security games also appear naturally at higher network layers as well as over internetworks. A popular line of work focuses on the problem of intrusion detection [9]. Again the attacker-defender model proves its usefulness, such as in a simple zero-sum stochastic game designed by Alpcan and Başar [2], and later by Zhu and Başar [151] with more realistic game models. This problem is also studied in collaborative settings where multiple intrusion detection systems (IDSs) cooperate, forming defenders-only games. For example, while Zhou et al. [150] consider rational IDSs being malicious,

Zhu et al. [152] focus on the problem of *free-riding*, i.e., some IDS might be lazy and does not contribute. Another typical ground for game modelling is the issue of Distributed Denial of Service (DDoS), which often involves a large body of Internet nodes. Spyridopoulos et al. [138] and Jafarian et al. [74] study different defense mechanisms against DDoS attacks under attacker-defender game model, as well as provide simulation results to strengthen their results. Meanwhile, Yan and Eidenbenz [148] investigate economics incentives for ISPs to collaborate in defending DDoS attacks, under defenders-only model.

1.4.2 Security Investment

Another natural place for game-theoretic modelling is in the organisational decision-making process of strategic investment in security. This problem is made interesting by the dependency of such process on not only the potential attackers, but also on other relevant parties in the environment, such as markets, alliances, competitors, etc.

A well-known game model for studying security investment is the *interdependent security* game (IDS) proposed by Heal and Kunreuther [84]. This is a defenders-only game among entities whose security are tightly linked to each others, e.g., security of airports against terrorists. Entities have an option to invest in a certain mechanism with interdependent effects in security. Alternatively, they may ignore it as freeriders, or because it becomes ineffective due to the lack of deployments by others. Different contexts for IDS games have been proposed, each with a distinct type of security interdependency. Grossklags et al. [65, 66] consider several models, including *weakest link*, *best shot*, and *total effort*: a player's security is determined by the smallest, largest, or total investment by all players, respectively. There are also models that consider *negative externalities*, i.e., deployments that have negative security effects on other players, such as by Hausken [69] and Miura-Ko et al. [98]. Several mechanisms have been proposed toward improving security and social utilities, e.g., using insurance [65, 66, 109, 111], auditing [22], information sharing [64, 109]. Further, Laszka et al. [86] provide a comprehensive survey of existing research on IDS games.

There are other topics in security investment which also benefit from game-theoretic analysis. A notable example is sharing of security information between firms (with independent security) as part of strategic security investment. Several works [54, 64, 71] of this type model "one-shot" games in which players' strategies are either investment or information sharing, or both. Others consider multi-stage games [71, 89] where information sharing may occur at one of the stages. In contrast to the above, another emerging line of research exploits the attacker-defender theme, notably the FlipIt game proposed by van Dijk et al. [141] that models strategic security plans against *advanced*

persistent threats (APTs) in taking over the control of a resource. Several follow-ups on FlipIt have appeared, e.g., an extension of the defender's strategies by Pham and Cid [114], an extension to multiple resources by Laszka et al. [87], as well as several empirical studies by Reitter et al. [124] and Nochenson et al. [106]. Last but not least, the growth of cloud and distributed computing also suggest a need for studies on security of outsourcing against dishonest contractors. Most of existing results come from cryptographic research, e.g., verifiable computation with fully-homomorphic encryption [55], probabilistically checkable proof [131, 133]. There are however emerging game-theoretic studies, such as by Belenkiy et al. [14], Nix and Kantarcioglu [104] which apply *principal-agent* model [58, ch.7] to incentivise honesty among contractors. Khouzani et al. [80] pursue this approach further to consider the problem of collusion between contractors.

1.5 Outline of Thesis Contributions

This section summarises the content of the remaining of the thesis, as well as explaining my shares of contribution for each of the works mentioned. There are four independent pieces of research, as presented in the next four consecutive chapters. The thesis ends with a conclusion chapter, along with an appendix. The main chapters are summarised below.

Test It Before Flipping It: Security Assessment Games

Strategic security investments are an important part in organisational decision making, due to the need for protection of information systems. Following a break to RSA [40], Dijk et al. [141] employ the notion of *games of timing* in economics [21] to propose a game called FlipIt that captures organisational strategies for security investment against an advanced persistent threat (APT) adversary. APTs are growing threats that cause detrimental effects to critical infrastructures [20]. In particular, FlipIt is a two-person game whose gameplay starts at zero time, and flows indefinitely. During the game play, players (attacker, defender) must decide when to make a move (usually system reset) to take over control of a single resource (e.g. computer system), the action of which incurs a cost, but which yields benefits from controlling the resource for the amount of time until it is taken by the opponent. In their paper, Dijk et al. consider several different types of moving strategies, including non-adaptive plans of moves that are decided before the gameplay begins, such as periodic, randomised strategies, as well as adaptive moves, i.e., moves decided during the gameplay based on observed history.

Our work is an extension to FlipIt by enriching further the types of moves that can be

made during the gameplay. We take on the argument that system resets are not enough in optimising organisational return on security investment, and that security assessment should also be taken into the strategies. To do so, we introduce the strategies called *state checking*, which allows a player to check if it is in control of the resource, and only make a move if not. We perform analysis to compare between this and the blind move strategies proposed by Dijk et al., with restriction to periodic type of strategies. Our analysis yields conditions under which one type of strategy is preferred over another, and vice versa. Further, we introduce another type move, called *hardening*, which allows the defender not just to check then move, but also patch the system vulnerability (after being taken by the attacker), so that later on it becomes harder for the attacker to take over. Our analysis of hardening strategies give further insights on how enterprises should spend on deep investigation to fix security problems of their infrastructures in the event of breaches.

This work is published in [115], in which all authors contribute equally. Particularly, I participate in seeking the research problem, proposing and revising the model, performing the analysis, as well as presenting the work in text format.

Strategic Information Sharing in Competitive Environments

In an inter-connected world where no single organisation can really stand alone and survive, there is a new trend of cooperation in which businesses do not just selfishly spend on protecting their information assets, but also gathering and exchanging security intelligence in helping themselves becoming more secure. The establishment of the US "National Coordinating Center for Communications (NCC)" as the "Information Sharing and Analysis Center (ISAC)" for telecommunications [108] is a sign of this trend. Meanwhile, many investigations on information sharing in general [52, 136, 143], as well as that specifically for security intelligence [53, 64, 70, 71] have been made by academics, hoping to devise a effective and efficient system for information exchange. Most of these works consider the problem of how firms should invest in securing their systems, as well as how to share information to others.

Different from previous works, we consider the question of how much firms should invest security research (e.g., Google Project Zero [50]), i.e., gaining information about the security of their platforms, alongside sharing the research results. Also, we model information as the number of vulnerabilities discovered instead of a value within [0,1] as in other works. We consider a two-stage game played between two competing firms in the same market who need to consider trade-offs between competition and sharing information for mutual goods. Our analysis of the equilibrium points indicates insights about firms' expected trends of decisions. Further, we introduce the notion of a me-

diator, and illustrate its effect on the firms' behaviours, which surprisingly can bring positive as well as negative outcomes to the individual firms, as well as social welfare.

This work is published in [81], in which all authors contribute fairly equally. In particular, I contribute mainly on designing/revising the model as well as in the analysis, with less efforts on presentation of the research results.

Optimal Contracts for Outsourced Computations

The idea of outsourcing complex computation tasks has been proposed and implemented in a variety of applications. These include search for extra-terrestrial life (*SETI@Home*), investigation of protein folding and computational drug design (*Folding@Home* and *Rosetta@home*). Businesses from different sections including finance, energy infrastructure, mining and commodities transport, technology and innovation [107] have also realised the benefits of outsourcing (data, computation, etc.) and "moving to the cloud". In all of these scenarios, there is a concern for the outsourcer (client) about the correctness of the returned results. With efforts from the cryptography community, many mechanisms have been proposed to address this concern, coming under the name *verifiable computation*, e.g., homomorphic encryption [56], Probabilistically Checkable Proofs (PCPs) [132,134], and then better schemes such as Pepper [131] and Ginger [133]. However, these mechanisms are still practically infeasible, due to the complexity of the computation process. From another route, there are economic studies that aim at forming principal-agent contracts that rationally discourage workers/agents from being dishonest.

Motivated by previous works that employ principal-agent contracts, we make an attempt to design more rigorous and general contracts for outsourcing, which are optimal in terms of outsourcing expense. Our contract design utilise auditing, reward, punishment in attracting workers, whilst at the same time encouraging their honesty. Further, we take into account several obstacles in reality not addressed by previous works, such as limited budget for expenditure, limited ability to punish, and limited capacity for auditing. In addition, we consider the problem of information leakage that gives workers extra advantages in cheating, as well as propose *bounty hunter* scheme to deter collusions among workers. Our results are presented as optimal contract parameters and expected cost, given different conditions of the environment parameters.

This work is published in [116] and [80], in which all authors contribute fairly equally. In particular, I contribute equally with others on designing and revising the game model, discussion of results, and presentation of the works. I am also responsible for all programming tasks that form the statements of the main theorems.

Rational Security for Unauthenticated Communication

Due to physical and geographical separation, secure communication is vital in our internetworked world. Various aspects of security have been established for this need, e.g., confidentiality, integrity, non-repudiation. Security services that can offer these aspects often operate on top of a mechanism that guarantees authenticity/correct identification of communicating parties. However, authentication in general is not an easy task. Although there exist effective authentication mechanisms such as Public Key Infrastructure (PKI) or Web of Trust (WoT), they are not efficient enough to be applicable to the vast majority of everyday connections such as news surfing or peer-to-peer file sharing. Quasi-authentication methods such as *leap-of-faith* have been proposed, but only work well in small and static environments, and hence is less scalable. Barak et al., on the other hand, consider the question: "What security can be achieved in the presence of an attacker, without authentication?" [10]. They give an answer in the form of a relaxed notion of security, which can easily be achieved using secure multiparty computation (SMPC) and digital signatures. Such relaxation states that, the most an adversary can do is to sequentially impersonate communicating endpoints. In other words, it cannot concurrently communicate with two or more endpoints and use information obtained from one side to talk with another. This strictly minimises the capability of an adversary as a man-in-the-middle attacker.

Inspired by this work, we also consider the same question, and arrive at the same notion of relaxation for security. However, we realise that the solution provided in [10] might be practically unfavourable. In particular, the use of SMPC may cause unbearable overhead in computation. Also, it is not compatible with modules unaware of SMPC, as these modules must be heavily reformatted in order to participate in the protocol execution. Motivated by this, we approach the question and the prescribed notion of security from a game-theoretic viewpoint to look for a more practical solution. We start by proposing formalisation of the notion of security, called *online-man-in-the-middle* (oMitM), with focus on query-response protocol, the most primal and common form of Internet communication nowadays. Then, we develop a game-theoretic model to study the equilibrium conditions under which a rational adversary would not violate oMitM security. The game is played between the adversary and the communicating endpoints. Finally, we develop a cryptographic protocol for the endpoints, so as to force the adversary into playing the prescribed game, and thus being encouraged to conform with oMitM security.

This work has not been published, and is fully a work of my own.

Chapter 2

Test It Before Flipping It: Security Assessment Games

Security assessments are an integral part of organisations' strategies for protecting their digital assets and critical IT infrastructure. In this chapter we propose a game-theoretic modelling of a particular form of security assessment – one which addresses the question "are we compromised?". We do so by extending FlipIt, a game model recently proposed by van Dijk et al. [141], which itself can be used to model the interaction between defenders and attackers under the Advanced Persistent Threat (APT) scenario. Our extension gives players the option to "test" the state of the game before making a move. This allows one to study the scenario in which organisations have the option to perform periodic security assessments of such nature, and the benefits they may bring.

2.1 Introduction

The protection of digital assets and critical IT infrastructure is an ever-growing concern for individuals, companies and nations. Information security is now a priority area for investment, given the growing threats from hackers, competitors, organised criminal gangs and enemy nation-states, and the potential for loss of privacy and revenue, negative reputational impact and effects in public welfare. In addition to direct investment in suitable and robust IT infrastructure, the performance of frequent security assessments is also considered an important component of the defense strategy against cyber-attacks. A security assessment is the process of determining how effectively an entity being assessed meets specific security objectives [100]. A common method of assessment is *penetration testing*, where security professionals target the network and other IT resources, to try to identify and verify any vulnerabilities found. Popular

penetration testing methodologies and frameworks work by essentially *mimicking* the popular forms of attack used by hackers.

The nature of cyber-attacks has however been steadily changing in recent years. While previously the typical threats were *script kiddies*, more interested in defacing websites for fun and pride, attacks motivated by financial gains are increasingly becoming more prevalent. Particularly in the corporate and government spheres, the threat of espionage and theft of intellectual property and state secrets are growing causes of concern. With these goals in mind, the methods used by attackers have also evolved. A form of attack that has received much attention recently are the so-called *Advanced Persistent Threats* (APT), which can often be seen as a signal of international cyber warfare [20]. The premises in this form of attack are that IT networks and systems *are* vulnerable, and therefore can be compromised by adversaries with enough resources and motivation; furthermore, attacks are stealthy in nature [35, 142], and adversaries can remain in control of the network and systems for long periods without detection. Recent examples of cyber attacks that fit this profile are the security breach at RSA Data Security [40], and the Stuxnet [85] worm infection of Iranian systems.

These developments should in turn motivate a reflection on whether current methods of security assessment remain sound under the changing nature of attacks. A security assessment is typically seen to be trying to answer the question "are we vulnerable?" (and if so, how can we fix it?). Under APT's premise, the answer for this question is certainly "yes". Thus a security assessment needs also to address the question "are we compromised?", and organisations need to consider cost-effective ways in which they can can regain control of their IT assets if the answer is positive. This current gap should certainly be the cause of concern for professionals involved in the security of highly-targeted organisations.[1]

In this chapter we propose a simple game-theoretic modelling of this form of security assessment, and study its application in two-player security games. Our model extends the recently proposed game FlipIt by van Diijk et al. [141], which itself can be used to model the interaction between defenders and attackers under the APT scenario. Our extension gives players the option to "test" the state of the game (i.e. answer the question "are we compromised"). This allows one to study the scenario in which organisations have the option of performing periodic security assessments of such nature, and the benefits they may bring. In particular, how these assessments can fit into an organisation's security investment strategy. Proposals of models for security

[1] In fact these points were emphatically argued in a recent testimony before the U.S.-China Economic and Security Review Commission Hearing on "Developments in China's Cyber and Nuclear Capabilities", where one of the participants stressed the need of periodic security assessments of the latter nature [12].

investment and security testing have appeared before in the literature (e.g. [23,24,63]); here we leverage on the elegance of FlipIt to investigate strategies for the application of this form of security assessment.

This chapter is organised as follows. We start with a review of related literature in Section 2.2. In Section 2.3 we describe the game FlipIt. In Section 2.4 we propose our extension to the game, by introducing the option of a security assessment which discloses the state of the game. We study further extensions in Sections 2.6 and 2.5, which address the probabilistic effectiveness of security assessment, as well as the ability to harden security over time. We finish with a conclusion in Section 2.7. For reader's convenience, we summarise the important notations in [table]

2.2 Related Work

The idea of a non-cooperative game in which players decide when to make a move appears long before the proposal of FlipIt, and is referred to as "games of timing" [21]. A typical game of timing consists of two players with zero-sum utility. Each player has a pool of resource to spend within a time interval (usually [0,1]). The players must select strategies to spend their resources over time so as to gain the most from each other at the end of the interval. The game becomes interesting when there are trade-offs between early and late expenses of one's resource. For example, consider two shooters in a *duel* with limited resource (ammunitions) walking toward each other. While conserving shots for later would increase accuracy (as distance shortened), early fires would reduce the chance of being shot in the first place [106].

Following a detailed survey by Radzik [120], studies of games of timing consider several different categories. In one dimension, players' resources are separated between *discrete* and *non-discrete* types, that is, they either consist of finite number of indivisible amounts, or can be distributed arbitrarily over the time interval, respectively. From another perspective, games can also be categorised based on information conditions of the players. For example, Karlin [77] distinguishes between games in which players have complete information about each other's expenditure over time, and otherwise games with limited information.

Applying to attacker-defender interaction in information security, van Dijk et al. [141] propose FlipIt with as a game of timing with distinguishable features. In FlipIt, the time interval is not bounded, but rather from 0 to ∞. This is to reflect the persistence of the APT attacker. Secondly, players' resources in FlipIt are costly (and hence unlimited), making it a non-zero sum game. FlipIt also belongs to the class of games with limited information. For example, non-adaptive games give no information

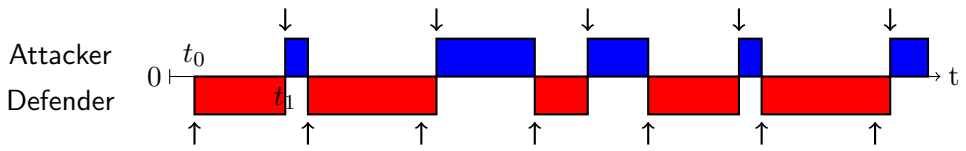


Figure 2.1: An example of FlipIt(P, P) game with periodic strategies with defender's phase t_0 and attacker's phase t_1. Each arrow indicates a flip to take over control of the resource.

to the players.

Our contributions. Our work is the first extension of FlipIt following the original proposal of the game. We focus on enriching the defense strategies, hoping to gather interesting insights for realistic security strategies. As explained and motivated in Section 2.3, we pick the FlipIt game with periodic moves as the basis for our extension. We contrast the strategies of blind-reset and check-then-reset of information systems. We also consider the investment strategies for hardening security when a breach occurs. Our theoretical results are then explained in executive language as emphases and recommendations for real-world security tactics.

2.3 FlipIt: The Game

Proposed by van Dijk et al. [141] in response to a data breach against RSA Security [40], the original FlipIt games capture the battle between a defender and an advanced persistent threat (APT) attacker for the control of a resource. The game is modelled over infinite time, in which a player makes a move to gain control of the resource; it remains in this state until the opponent makes its own move to take over. This control-alternating process repeats infinitely as time passes, and the utility of each player is determined by the total/average amount of time it controls the resource, as well as the cost required to take over the resource from its opponent. An example of the game play is illustrated in Figure 2.1.

Formally, we consider FlipIt as a strategic-form game $\langle N, \{A_i\}, \{u_i\}\rangle$ where the set of players $N = \{0, 1\}$ contains the defender and the attacker, denoted by player 0 and 1, respectively. FlipIt models a situation starting from some time moment $t = 0$ and is continuously indefinite, during which the control of a resource R is being alternated between the attacker and the defender. Particularly, let $C_i(t)$ be 1 if player i controls the resource at time t, and 0 otherwise, where $C_0(0) = 1$, i.e., the game starts with the defender controlling the resource. A player can claim control of the resource by making a *move*, or *flip*. For instance, if the defender moves at time t, then $C_0(t) = 1$; similarly,

we have $C_0(t') = 0$ if the attacker moves at time t'. This allows the total control time of player i until time t to be computed as

$$G_i(t) = \int_0^t C_i(t)dt.$$

Denote player i's number of moves until time t by $n_i(t)$, and the constant cost for each move by k_i; then the *net benefit* of player i is given by

$$B_i(t) = G_i(t) - n_i(t)k_i.$$

Alternatively, since the game continues indefinitely, a player's utility can be represented by its *average benefit* per unit time:

$$u_i(t) = \frac{B_i(t)}{t} = \frac{G_i(t)}{t} - \frac{n_i(t)}{t}k_i = \gamma_i(t) - \alpha_i(t)k_i.$$

We call $\gamma_i(t)$ and $\alpha_i(t)$ the *average gain rate* and the *average move rate* of player i up to time t, respectively. One may further assume that the functions $\gamma_i(t)$ and $\alpha_i(t)$ converge to the values γ_i and α_i, respectively, as $t \to \infty$. We can then conveniently represent player i's utility without the time dimension as simply

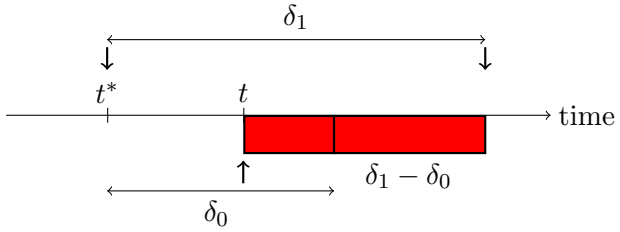
$$u_i = \lim_{t\to\infty} u_i(t) = \gamma_i - \alpha_i k_i. \tag{2.1}$$

What remains to be modelled are γ_i and α_i, which strongly depend on how the players strategically act in the game. While the original work [141] discusses several types of strategies for each player, we focus only on the so-called *periodic strategies with random phase* (Figure 2.1), which is the main tool in our research. In this case, we assume that before start, each player chooses a rate $\alpha_i > 0$ so that as the game progresses, player i moves at rate α_i, i.e., after every $\delta_i = 1/\alpha_i$ units of time. Furthermore, player i does not start moving immediately at $t = 0$, but selects uniformly at random a starting point in the interval $[0, \delta_i]$; this is called *phase*. The use of random phase illustrates the fact that the defender and the attacker in reality only know each other's flipping period, not the exact flipping time. While players cannot control their phases, their game action is determined by the chosen move rates, i.e, α_0 (defender) and α_1 (attacker). For convenience, we denote the strategy space for periodic moving strategies for both players as

$$A_0 = A_1 = P = \{P_\alpha | \alpha > 0\}.$$

Assume that players move periodically following a strategy/action profile $(P_{\alpha_0}, P_{\alpha_1}) \in A_0 \times A_1$, their average benefit, or utilities, can be computed in the following two cases:

- $\alpha_0 \geq \alpha_1$: since $\delta_0 = 1/\alpha_0 \leq 1/\alpha_1 = \delta_1$, let $r = \alpha_1/\alpha_0 = \delta_0/\delta_1 \in [0, 1]$. Let t^* be a moment when an attacker's move/flip occurs, then the next attacker's move is at time $t^* + \delta_1$. Since the defender moves every period of δ_0 and that $\delta_0 \leq \delta_1$, it must move exactly once at some $t \in [t^*, t^* + \delta_0] \subset [t^*, t^* + \delta_1]$, as seen below:



  Due to the assumption of random phase, t is uniformly distributed within $[t^*, t^* + \delta_0]$, yielding a gain $(t^* + \delta_0 - t) + (\delta_1 - \delta_0) = t^* + \delta_1 - t$, and thus the defender's expected control time within $[t^*, t^* + \delta_0]$ is:

$$G_0^* = \int_{t^*}^{t^*+\delta_0} \frac{t^* + \delta_1 - t}{\delta_0} dt = \delta_1 - \frac{\delta_0}{2} = \delta_1(1 - \frac{r}{2}). \tag{2.2}$$

  This implies that the defender's average gain is $\gamma_0 = G_0^*/\delta_1 = 1 - r/2$; it also means that the attacker's average gain is $\gamma_1 = 1 - \gamma_0 = r/2$. Therefore, we have the players' utilities as

$$u_0(P_{\alpha_0}, P_{\alpha_1}) = 1 - \frac{r}{2} - \alpha_0 k_0 = 1 - \frac{\alpha_1}{2\alpha_0} - \alpha_0 k_0,$$
$$u_1(P_{\alpha_0}, P_{\alpha_1}) = \frac{r}{2} - \alpha_1 k_0 = \frac{\alpha_1}{2\alpha_0} - \alpha_1 k_1.$$

- $\alpha_0 \leq \alpha_1$: similar analysis gives the following

$$u_0(P_{\alpha_0}, P_{\alpha_1}) = \frac{r}{2} - \alpha_0 k_0 = \frac{\alpha_0}{2\alpha_1} - \alpha_0 k_0,$$
$$u_1(P_{\alpha_0}, P_{\alpha_1}) = 1 - \frac{r}{2} - \alpha_1 k_0 = 1 - \frac{\alpha_0}{2\alpha_1} - \alpha_1 k_1.$$

We note that when a player has lost the control due to the opponent's move, it does not immediately move to regain it but rather needs to wait for its periodic move. This is because moves are presumably "stealthy", and neither player knows at any time who is controlling the resource. In addition to the periodic move scenario, [141] also studies strategies involving randomised moves, as well as adaptive strategies based on

the opponent's past moves. Although we do not consider these here, we note that the modelling presented in this chapter may be similarly applied to other scenarios discussed in [141]. For reader's convenience, we summarise the notion of FlipIt game with periodic strategies, as well as important notations below.

Definition 2.1. *Let k_0 and k_1 are per-move cost for player 0 and 1, respectively. A* FlipIt *game is a strategic-form game $\langle N, \{A_i\}, \{u_i\}\rangle$, denoted as* FlipIt$(A_0, A_1)$ *with utility functions as in* (2.1)*. A* FlipIt *game with periodic strategies is of the form* FlipIt(P, P) *where $P = \{P_\alpha | \alpha > 0\}$, such that for every action profile $(P_{\alpha_0}, P_{\alpha_1}) \in P^2$ players' utilities are:*

$$u_i(P_{\alpha_i}, P_{\alpha_{-i}}) = \begin{cases} 1 - \dfrac{\alpha_{-i}}{2\alpha_i} - \alpha_i k_i & \text{if } \alpha_i \geq \alpha_{-i}, \text{ or} \\ \dfrac{\alpha_i}{2\alpha_{-i}} - \alpha_i k_i & \text{otherwise.} \end{cases}$$

Remark. We choose FlipIt with periodic strategies to base our work on due to several reasons. FlipIt is a simple, though elegant, model of real-world IT security attacker-defender interaction that emphasises the persistence aspect of the attacker. Also, strategies for organisational security are often determined in the very early phase of the business, and they are normally deterministic (quarterly assessments, periodic guard patrolling, etc.) rather than being oblivious and temporary [101]. This suggests us to consider periodic strategies, as it would be most applicable to reality. In another aspect, the original model assumes complete "stealthiness" of players' moves, since no player obtain any information during the game. As will be described, our extension proposes state-checking, indicating that a player might obtain useful information while playing and thus might adapt its strategy. However, we do not consider the advantage of adaptive strategies. Instead, we stick to a comparable model in order to reveal more accurately the differences our extensions make to the original model. Finally, our choice takes into account simplicity of modelling to enable effective analysis and useful insights into the problem.

2.4 Test It before Flipping it

The original FlipIt game proposed by van Dijk [141] models different types of strategies for a player to regain control of a resource (i.e. to move) based on some pre-defined or on-the-fly tactics, which however possess some limitations. In particular, a player may waste many moves if they happen while it is still controlling the resource. This becomes more serious if its periodic movement is significantly faster than the opponent's. Even if a move really serves its purpose, i.e., to regain control, it may still be an "almost"

Table 2.1: List of main notations

parameter	definition
α_i	periodic moving/state-checking rate of player i
δ_i	a period between two consecutive move/state check of player i
k_i	cost of each move/flip to player i
c_i	cost of each state check to player i
h_i	cost of each security hardening to player i
P_{α_i}	a periodic moving strategy with rate α_i of player i
S_{α_i}	a periodic state-checking strategy with rate α_i of player i
H_{α_i,h_i}	a periodic state-checking strategy with rate α_i and hardening cost h_i of player i
p	probability that a defender's state check succeeds in finding a breach
f	the function that updates the attack cost after each hardening
μ	the maximum increase in attack cost after each hardening
λ	the effectiveness of the hardening process

waste. This happens, for example, when the opponent's move is immediately (but coincidently) after such a move, rendering it ineffective.

Rather than blindly moving, an interesting question is whether knowing the state of control would be more beneficial to a player. In terms of information security assessment, this can be represented by the question "are we compromised?". The intuition behind this addition is rather simple. Knowing the state of control would prevent a waste move while the resource is still at hand. Also, even though it may not prevent an "almost" waste, it may suggest a timely response to a lost of control. This, of course, depends on how regularly the knowledge of the control state is updated.

To model such situations, we introduce a new class of strategies to FlipIt, namely the *state checking* strategies, as illustrated in Figure 2.2. As opposed to the ability to move/flip, a player is now able to check the game state, and then move/flip if necessary. In particular, we consider a strategy class $S = \{S_\alpha | \alpha > 0\}$ such that, given a strategy $S_\alpha \in S$, with $\delta = 1/\alpha$, player i may:

- perform a periodic state checking with period δ and *state-checking cost* c_i, with the first check occurring at a uniformly random time phase, i.e., within $[0, \delta]$;
- if a state check indicates a loss of control, immediately perform a move/flip (at

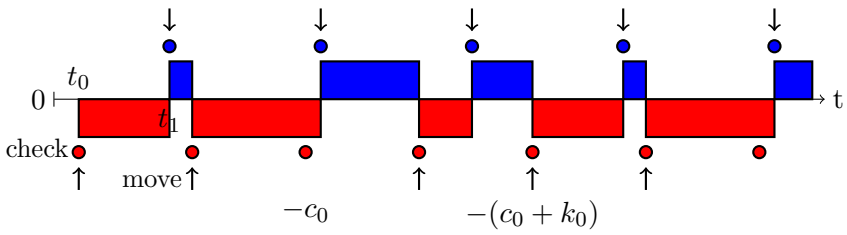


Figure 2.2: An example of a $\mathsf{FlipIt}(S,S)$ game with period state checking strategies.

cost k_i) to regain its control.

In addition to the original game $\mathsf{FlipIt}(P,P)$, several games might be introduced given S, for example $\mathsf{FlipIt}(S,P)$, $\mathsf{FlipIt}(S\cup P,P)$, and $\mathsf{FlipIt}(S\cup P,S\cup P)$. To study such games, it is important to notice that in all cases, the expected control time for each player can be formulated in the same way as that in $\mathsf{FlipIt}(P,P)$, using only δ_0 (or α_0) and δ_1 (or α_1). Indeed, at a time t, if a player $-i$ is occupying the resource, a blind move action (specified by P_{α_i}) or a check-then-move action (specified by S_{α_i}) would yield the same effect, i.e., allowing player i to regain control. In contrast, if i is in control of the resource, then neither action yields any change. As this happens independently of the opponent's strategy, P_{α_i} and $S_{\alpha_i'}$ would yield the same average control time as long as $\alpha_i=\alpha_i'$.

The main difference between P_{α_i} and S_{α_i} are in the cost of operating. With a strategy S_{α_i}, the average state checking cost for player i is $\alpha_i c_i$. There is however the average cost of moving, which is much less than in periodic moving, since a move is only made when a loss of control is confirmed. This means that player i's number of moves is at most player $-i$'s number of moves, i.e., $\min(\alpha_i,\alpha_{-i})$. This allows the construction of its utility to become

$$u_i(S_{\alpha_i},P_{\alpha_{-i}}\vee S_{\alpha_{-i}})=\begin{cases}1-\frac{\alpha_{-i}}{2\alpha_i}-c_i\alpha_i-k_i\alpha_{-i} & \text{if } \alpha_i\geq\alpha_{-i}, \text{ or}\\ \frac{\alpha_i}{2\alpha_{-i}}-c_i\alpha_i-k_i\alpha_i & \text{if } \alpha_i<\alpha_{-i}.\end{cases} \tag{2.3}$$

Given this new type of strategies S, a natural approach is to compare between S and P, that is, in which situations one is preferred over the other. The following proposition provides such comparison based on the relation between the costs of moving and state checking.

Proposition 2.1. *In the game* $\mathsf{FlipIt}(P\cup S,\ P\cup S)$*, if* $c_i\leq k_i/4$*, player* i *does not prefer periodic moving. Otherwise, when* $c_i\geq k_i$ *player* i *does not prefer state checking.*

Proof. This proposition is a special case of Proposition 2.2, when $p=1$. □

Corollary 2.1. *Consider the game* $\mathsf{FlipIt}(P \cup S, P \cup S)$ *with* $k_i/4 < c_i < k_i$. *Player* i *prefers a state checking strategy if and only if* $\alpha_{1-i} \leq \frac{2(\sqrt{k_i}-\sqrt{c_i})^2}{k_i^2}$.

Discussions. The above results point out that when the cost of checking is sufficiently low, i.e., at most a quarter of the moving cost, it is always worth performing a check-then-move strategy. Indeed, as a low checking cost suggests a frequent checking schedule, a player is more closely up-to-date with its state of control of the resource. This helps the player to improve its expected control time, while keeping the moving cost at a reasonable level by eliminating wasted moves. Conversely, it is also intuitively clear that when the cost of checking exceeds that of moving, it is unreasonable to perform check-then-move. Interestingly, Corollary 2.1 also indicates that, when the two cost are comparable, the best response for the opponent playing too fast (exceeding threshold $t = \frac{2(\sqrt{k_i}-\sqrt{c_i})^2}{k_i^2}$) is to either simply move at every step or not play at all, because at every step it is likely that without state checking the player is aware of its loss of control of the resource. Because $\partial t/\partial c < 0$ and $\partial t/\partial k > 0$, such threshold also agrees with the fact that state checking is more preferable when the checking cost c_i is low and the moving cost k_i is high. In the realm of information security, many situations may suggest that state checking strategies indeed outperform their moving counterparts. Consider an information system as the resource; the defender's act of moving/flipping is often expensive, as it might involve resets and restores of the system. This becomes more serious for large organisations, or those that require uninterrupted, real-time system availability and reliability, such as e-commerce, large computing facilities. On the other hand, checking for successful take-over of the system might be significantly cheaper and thus can be performed frequently, using intrusion detection systems (IDSs), auditing schemes, logging, etc. In such cases, it is recommended that funds are allocated for more frequent auditing of the system security to maximise the organisational benefit from the information system.

2.5 Dealing with Complex Systems

In this section, we study a different extension to the previous model to capture situations in which the control of a resource might be difficult to measure, and that state checking might be inaccurate. This disregards an inherent but hidden assumption that with a cost c_i, player i can always determine who is in control of the resource. Again, it addresses another important issue with organisational information security by exacerbating the question "are we compromised?" by "how certain are we whether we are compromised?". An answer to such question reflects not just how often security should

be assessed, but also how the assessment should be done. We extend the previous state-checking model with a probability p that the state check succeeds in determining a loss of control, applied to the defender only. The reason for such bias is obvious: while the defender must examine every component of its system as a mean of state checking, the attacker only needs to consider what it has previously compromised, which normally happens with certainty. For the sake of analysis, we explicitly make two assumptions:

A1. There exists no false positive in state checking, i.e., no false alarm on attack exists. In other words, the defender would only consider that a breach occurs if an evidence is found in favour of it. [2]

A2. Once a false negative occurs, it will persist until the attacker's next interaction with the resource, i.e., either via a state check, or a move/flip. [3]

Based on these assumptions, we may reformulate the defender's utility functions from what is given in (2.3), with the help of Lemma 2.1.

Lemma 2.1. *Consider the game* $\mathsf{FlipIt}(S, P \cup S)$ *in which the defender's state check succeeds with probability* p*, along with assumption* ***A1*** *and* ***A2****. Then the defender's utility function is*

$$u_0(S_{\alpha_0}, P_{\alpha_1} \vee S_{\alpha_1}) = \begin{cases} p\left(1 - \dfrac{\alpha_1}{2\alpha_0}\right) - c_0\alpha_0 - pk_0\alpha_1 & \text{if } \alpha_0 \geq \alpha_1, \text{ or} \\ p\left(\dfrac{\alpha_0}{2\alpha_1}\right) - c_0\alpha_0 - pk_0\alpha_0 & \text{otherwise.} \end{cases} \tag{2.4}$$

Proof. Like in most other proofs, we consider two cases. The first case is $\alpha_0 \geq \alpha_1$ which implies $\delta_0 \leq \delta_1$. In that case for every time interval $[t, t + \delta_1]$ between two consecutive attacker's moves/state-checks, the defender's utility given that it performs the state check at time $t_0 \in [t, t + \delta_0]$ is $p(t + \delta_1 - t_0)$. For $t_0 > t + \delta_0$ it means that the defender has already done a check before that for the same attack, and it has not succeeded, then the check at t_0 also fails (due to assumption **A2**). Thus the defender's average

[2] Although false positives are normal in intrusion detection (especially when automated), our reasoning is that, for the purpose of system reset, a careful investigation should be done both automatically and manually to verify the alleged breach. For simplicity we therefore neglect the possibility of false positive.

[3] In defending this assumption, if the attacker's move has been stealthy and that the defender fails to detect it, then if the attacker takes no further action, the defender receives no useful information to have a better chance in rediscovering the breach. Although in reality the state-checking process may be probabilistic and would succeed without further information, we assume otherwise to take into account the worst-case scenario.

control rate is

$$\frac{1}{\delta_1\delta_0}\int_t^{t+\delta_0} p(t+\delta_1-t_0)dt = p\left(1-\frac{\alpha_1}{2\alpha_0}\right)$$

For every attacker's move there is a probability p that it is discovered, which is the only situation the defender might make a move (due to **A1**), and thus the defender's cost due to moving/flipping is $pk_0\alpha_1$. For the case $\alpha_0 < \alpha_1$, which implies $\delta_0 > \delta_1$, we consider interval $[t_0, t_0+\delta_0]$ between two consecutive defender' state-checks. We notice right before t_0 the resource is controlled by the attacker, and if the state check succeeds at t_0 then the defender's average control time is $\frac{1}{2\alpha_1}$, and otherwise it is 0. However, every state check independently succeeds with probability p, and hence the defender's average control rate is $\frac{p\alpha_0}{2\alpha_1}$. Because each check succeeds with probability p, the defender's move rate is thus $\alpha_0 p$, yielding the cost $pk_0\alpha_0$. □

Similar to the its predecessor, with this model we are also interested in the conditions under which state checking is preferred to mere flipping, and vice versa. This concern is reflected in Proposition 2.2, which generalises the result given in Proposition 2.1, and thus emphasises a preference for strategies involving inexpensive state checking, i.e., equal to at most a $p/4$-fraction of the flipping cost.

Proposition 2.2. *Consider the game* $\mathsf{FlipIt}(P \cup S, P \cup S)$ *with the defender's utility* (2.4)*. Let* P_{α_1} *or* S_{α_1} *be the attacker's strategy. The defender prefers* S_{α_0} *over* P_{α_0} *if*

$$c_0 \le \frac{k_0 p}{4} \quad \textit{and} \quad \alpha_1 \ge \frac{1}{2k_0}\min\left(1, \left[\frac{2(1-p)}{p}\right]^2\right). \tag{2.5}$$

Proof. To compare between two classes of strategies, we first compute the best response function for each class. For simplicity of presentation, we use $\mathsf{BR}_0^P(\cdot) = \alpha_0$ to signify that the best periodic flipping response is with rate α_0, and similarly $\mathsf{BR}_0^S(\cdot) = \alpha_0$ for the best periodic state-checking response. We reuse the best response for periodic flipping strategies from the original model in [141] as follows:

$$\mathsf{BR}_0^P(P_{\alpha_1} \vee S_{\alpha_1}) = \begin{cases} 0, & \text{if } \alpha_1 > \overline{\alpha_1} \\ \left[0, \sqrt{\frac{\alpha_1}{2k_0}}\right], & \text{if } \alpha_1 = \overline{\alpha_1} \\ \sqrt{\frac{\alpha_1}{2k_0}}, & \text{if } \alpha_1 < \overline{\alpha_1} \end{cases} \tag{2.6}$$

where $\overline{\alpha_1} = \frac{1}{2k_0}$. For periodic state checking, we consider two cases:

- $\alpha_0 \geq \alpha_1$: from (2.4) we compute the best response of α_0 to α_1 as

$$\frac{\partial u_0(\alpha_0, \alpha_1)}{\partial \alpha_0} = \frac{p\alpha_1}{2\alpha_0^2} - c_0 = 0 \Leftrightarrow \alpha_0 = \mathsf{BR}_0^S(P_{\alpha_1} \vee S_{\alpha_1}) = \sqrt{\frac{p\alpha_1}{2c_0}} \geq \alpha_1$$

  This is thus valid only when $\alpha_1 \leq \frac{p}{2c_0}$.

- $\alpha_0 \leq \alpha_1$: since the derivative of u_0 with respect to α_0 is now

$$\frac{\partial u_0(\alpha_0, \alpha_1)}{\partial \alpha_0} = \frac{p}{2\alpha_1} - (c_0 + k_0 p)$$

  we can further divide this case to three sub-cases:

  - $\frac{p}{2\alpha_1} - (c_0 + k_0 p) > 0 \Leftrightarrow \alpha_1 < \frac{p}{2(c_0 + k_0 p)} : \mathsf{BR}_0^S(P_{\alpha_1} \vee S_{\alpha_1}) = \alpha_1$
  - $\frac{p}{2\alpha_1} - (c_0 + k_0 p) = 0 \Leftrightarrow \alpha_1 = \frac{p}{2(c_0 + k_0 p)} : \mathsf{BR}_0^S(P_{\alpha_1} \vee S_{\alpha_1}) \in \left[0, \frac{p}{2(c_0 + k_0 p)}\right]$
  - $\frac{p}{2\alpha_1} - (c_0 + k_0 p) < 0 \Leftrightarrow \alpha_1 > \frac{p}{2(c_0 + k_0 p)} : \mathsf{BR}_0^S(P_{\alpha_1} \vee S_{\alpha_1}) = 0$

We now combine these observations, this time by considering different values of α_1:

- $\alpha_1 \geq \frac{p}{2c_0}$: for $\alpha_0 \geq \alpha_1$, since $\sqrt{\frac{p\alpha_1}{2c_0}} \leq \alpha_1$, u_0 decreases, thus it maximises at $\alpha_0 = \alpha_1$. For $\alpha_0 \leq \alpha_1$, since $\alpha_1 \geq \frac{p}{2c_0} \geq \frac{p}{2(c_0 + k_0 p)}$, u_0 maximises at $\alpha_0 = 0$. Thus in overall, $\mathsf{BR}_0^S(P_{\alpha_1} \vee S_{\alpha_1}) = 0$.

- $\frac{p}{2(c_0 + k_0 p)} < \alpha_1 < \frac{p}{2c_0}$: we consider two cases

  - $\alpha_0 \leq \alpha_1$: since $\alpha_1 > \frac{p}{2(c_0 + k_0 p)}$, we have $\mathsf{BR}_0^S(P_{\alpha_1} \vee S_{\alpha_1}) = 0$ and $u_0 = 0$.
  - $\alpha_0 \geq \alpha_1$: since $\mathsf{BR}_0^S(P_{\alpha_1} \vee S_{\alpha_1}) = \sqrt{\frac{p\alpha_1}{2c_0}}$ we have the defender's utility

$$u_0 = p\left(1 - \frac{\alpha_1}{2\sqrt{\frac{p\alpha_1}{2c_0}}}\right) - c_0\sqrt{\frac{p\alpha_1}{2c_0}} - pk_0\alpha_1 = 0$$

$$\Leftrightarrow \overline{\alpha_1}^* = \frac{p}{c_0 + k_0 p + \sqrt{c_0(c_0 + 2k_0 p)}} \in \left[\frac{p}{2(c_0 + k_0 p)}, \frac{p}{2c_0}\right],$$

  so that $u_0 < 0$ (resp. $u_0 > 0$) when $\alpha_1 > \overline{\alpha_1}^*$ (resp. $\alpha_1 < \overline{\alpha_1}^*$).

Compare these two subcases we may conclude that

$$\mathsf{BR}_0^S(P_{\alpha_1} \vee S_{\alpha_1}) = \begin{cases} 0, & \text{if } \overline{\alpha_1}^* < \alpha_1 < \frac{p}{2c_0} \\ 0 \text{ or } \sqrt{\frac{p\alpha_1}{2c_0}}, & \text{if } \alpha_1 = \overline{\alpha_1}^* \\ \sqrt{\frac{p\alpha_1}{2c_0}}, & \text{if } \frac{p}{2(c_0+k_0p)} < \alpha_1 < \overline{\alpha_1}^* \end{cases}$$

- $\alpha_1 \leq \frac{p}{2(c_0+k_0p)}$: again, two sub-cases exist:
  - $\alpha_0 \geq \alpha_1$: u_0 maximises at $\alpha_0 = \sqrt{\frac{p\alpha_1}{2c_0}} > \alpha_1$
  - $\alpha_0 \leq \alpha_1$: u_0 maximises at $\alpha_0 = \alpha_1$

  Since u_0 is continuous over α_0, it thus maximises at $\sqrt{\frac{p\alpha_1}{2c_0}} = \mathsf{BR}_0^S(P_{\alpha_1} \vee S_{\alpha_1})$.

The above analysis concludes the defender's utility function for state checking as

$$\mathsf{BR}_0^S(P_{\alpha_1} \vee S_{\alpha_1}) = \begin{cases} 0, & \text{if } \alpha_1 > \overline{\alpha_1}^* \\ 0 \text{ or } \sqrt{\frac{p\alpha_1}{2k_0}}, & \text{if } \alpha_1 = \overline{\alpha_1}^* \\ \sqrt{\frac{p\alpha_1}{2k_0}}, & \text{if } \alpha_1 < \overline{\alpha_1}^* \end{cases} \tag{2.7}$$

where $\overline{\alpha_1}^* = \frac{p}{c_0+k_0p+\sqrt{c_0(c_0+2k_0p)}}$. To compare between $\mathsf{BR}_0^P(P_{\alpha_1} \vee S_{\alpha_1})$ and $\mathsf{BR}_0^S(P_{\alpha_1} \vee S_{\alpha_1})$ we first notice that since $c_0 \leq k_0p/4$, we also have $\overline{\alpha_1}^* \geq \overline{\alpha_1}$:

$$\begin{aligned} \overline{\alpha_1}^* - \overline{\alpha_1} &= \frac{p}{c_0 + k_0p + \sqrt{c_0(c_0+2k_0p)}} - \frac{1}{2k_0} \\ &= \frac{\sqrt{4c_0^2 + 4c_0k_0p + (k_0p)^2} - \sqrt{4c_0^2 + 8c_0k_0p}}{2k_0^2p} \\ &\geq \frac{\sqrt{4c_0^2 + 4c_0k_0p + 4c_0(k_0p)} - \sqrt{4c_0^2 + 8c_0k_0p}}{2k_0^2p} = 0 \end{aligned}$$

We are now ready to compare BR_0^P and BR_0^S for different choices of α_1:

- $\alpha_1 \geq \overline{\alpha_1}^*$: since $\mathsf{BR}_0^P(P_{\alpha_1} \vee S_{\alpha_1}) = 0$ and $\mathsf{BR}_0^S(P_{\alpha_1} \vee S_{\alpha_1}) = 0$ or $\sqrt{\frac{p\alpha_1}{2c_0}}$, both yielding 0 utility, and thus periodic state checking is as good as periodic flipping.
- $\overline{\alpha_1} \leq \alpha_1 < \overline{\alpha_1}^*$: since $\mathsf{BR}_0^P(P_{\alpha_1} \vee S_{\alpha_1}) = 0$, thus periodic flipping achieves at most 0 utility, hence periodic state checking is preferred.
- $\alpha_1 < \overline{\alpha_1}$: we compare the optimal utility between two types of strategies:

– $\mathsf{BR}_0^P(P_{\alpha_1} \vee S_{\alpha_1}) = \sqrt{\frac{\alpha_1}{2k_0}}$, the defender's utility is:

$$u_0 = 1 - \frac{\alpha_1}{2\alpha_0} - k_0\alpha_0 = 1 - \sqrt{2k_0\alpha_1}$$

– $\mathsf{BR}_0^S(P_{\alpha_1} \vee S_{\alpha_1}) = \sqrt{\frac{p\alpha_1}{2c_0}}$, the defender's utility is:

$$u_0^* = p\left(1 - \frac{\alpha_1}{2\alpha_0}\right) - c_0\alpha_0 - pk_0\alpha_1 = p - k_0p\alpha_1 - \sqrt{2c_0p\alpha_1}$$

We then study the condition under which $u_0^* \geq u_0$:

$$\begin{aligned} u_0^* - u_0 &= p - 1 + \sqrt{2\alpha_1}(\sqrt{k_0} - \sqrt{c_0p}) - k_0p\alpha_1 \\ &\geq p - 1 + \sqrt{2\alpha_1}(\sqrt{k_0} - \sqrt{k_0}\frac{p}{2}) - k_0p\alpha_1 \end{aligned} \tag{2.8}$$

The right-hand side of (2.8) is non-negative if and only if α_1 satisfies

$$\frac{1}{2k_0}\min\left(1, \left[\frac{2(1-p)}{p}\right]^2\right) \leq \alpha_1 \leq \frac{1}{2k_0}\max\left(1, \left[\frac{2(1-p)}{p}\right]^2\right)$$

Since we only consider $\alpha_1 < \overline{\alpha_1} = 1/(2k_0)$, we thus conclude that $u_0^* \geq u_0$, that is, periodic state checking is preferred, whenever α_1 satisfies (2.5).

□

Corollary 2.2. *Consider the game* $\mathsf{FlipIt}(P \cup S, P \cup S)$ *with the defender's utility function* (2.4)*. Let* $\overline{\alpha_1}$ *(resp.* $\overline{\alpha_1}^*$*) be the minimum value for the attacker's move rate* α_1 *to drop a periodic-moving (resp. state-checking) defender from the game. Then,* $\overline{\alpha_1}^* \geq \overline{\alpha_1}$ *if and only if* $c_0 \leq k_0p/4$*.*

Discussions. Proposition 2.2 points out a simple condition under which the defender would prefer state-checking over periodic flipping, involving only the costs of checking, flipping and the checking effectiveness p. It also emphasises in (2.5) that state-checking is only efficient against frequently active attackers. Otherwise, if the attacker infrequently interacts with the resource, then by assumption **A2** the defender receives little information to improve its chance of attack detection, thus periodic flipping would be more desirable even if it is excessively expensive. The need for $c_0 \leq k_0/4$ is further strengthened by Corollary 2.2 which addresses the situation when the attacker plays too fast, e.g., $\alpha_1 > 1/(2k_0)$, and periodic moving cannot afford for positive payoff, leading to the system being indefensible [141]. This issue becomes more realistic when the

attacker is given chances to perform state checking, since in the information security realm, the attacker's state checking can be inexpensive, e.g., reconnecting to backdoors, re-logging in with stolen passwords, etc. In this case, periodic state checking is more robust as they survive higher attack rates.

Another intrinsic part of Proposition 2.2 is its implication over what is the right cost for state checking. Indeed, flipping in security often involves procedures with high certainty (system reset, backup restores, failovers, etc.), hence their costs are normally determined rather than decided. In contrast, an organisation may choose to invest arbitrarily in administering its security, for example through guard patrolling, antivirus software, firewalls, etc., subject to how much it desires the situation to be in control. While the goal is to satisfy the condition $c_0 \leq pk_0/4$, it is hindered by an inherent constraint that p typically decreases/increases with c_0, that is, less efforts for state checking yields less certainty on its effectiveness. We study this issue by modelling the connection between c_0 and p, along with an environment parameter $v > 0$ specifying how effectively the amount c_0 might be spent. For example, this parameter may deteriorate as the resource becomes increasingly more sophisticated. On the other hand, it may increase with the skills of the team performing state checking. We model p as the function of c_0, parameterised by v in the following way

$$p_v(c_0) = 1 - \frac{1}{vc_0 + 1}. \tag{2.9}$$

It is not difficult to see that, by modelling the probability of successful state checking as in (2.9), the value $1/v$ represents the cost required for detection of attacks to succeed with a fair coin-flipping chance, i.e., 50%. Note that this does not mean state checking with cost $c_0 \leq 1/v$ can be replaced by "coin-flipping detection" of attacks, as it may violate assumption **A1** to create many false positives, and hence waste moves would become a credible threat to the net utility. We now analyse the threshold under which the cost for state checking suggests it to overpower merely periodic flipping strategies.

Corollary 2.3. *Consider the game* $\mathsf{FlipIt}(P \cup S, P \cup S)$ *with the defender's utility function* (2.4)*, where* p *satisfies* (2.9)*. Then, if* $c_0 \leq k_0/4 - 1/v$ *and* $\alpha_1 \geq \frac{1}{2k_0} \min\left[1, \frac{4}{(c_0 v)^2}\right]$*, it is better for the defender to perform periodic state checking.*

From the threshold for state-checking cost given in Corollary 2.3, we may also evaluate whether state checking is at all justifiable given specific characteristics of the environments. Indeed, if the productivity of information security is too low, i.e., $v \leq 4/k_0$, the use of state checking in most cases would not improve the overall utility, as too much cost is required to produce little benefit. This refers to situations when

there is a mismatch between the scope of the resource being administered, and of the team performing administration, which means either the resource is too complex, or the administration is immature. In turn, such situations may apply to fast-growing organisations with slower catching-up with technology as well as security evaluation. Another example is with small to medium-sized firms whose businesses strongly rely on information systems, as many of them would spend little research in foreseeing the non-trivial impact of low security administration to the net income.

In overall, Corollary 2.3 recommends firms not just about hiring an administration team with highest quality-price ratio, but also to spend their concerns on easing the administration of their resource. In reality, the latter can be accomplished in a variety of ways, such as removing redundant components, restructuring the system toward simplification, avoiding complicated dependencies using separation of duties, etc. Otherwise, even the most desirable administration team might still be insufficient for a positive return on investment.

2.6 Hardening Control over Time

Besides reactive measures such as state checking and moving, a proactive concern is on how to prevent losses of control from happening. In many cases this is more desirable because it is possible that consequences from attacks might have been overlooked, and thus it is better that attacks are prevented given the current realisation of potential losses. In the context of FlipIt, it may mean, for example, preventing a player from participating in the game, or to stop it after the game has run for some time. Following the analysis of the original FlipIt game, as well as those involving state checking strategies, it is not difficult to see that in order for a player to stop its opponent from participating in the game, it needs to play quick enough. Assume in the best case that state checking succeeds with probability $p = 1$, and based on the best response functions for periodic moving (2.6) and periodic state checking (2.7), player i should pick a strategy P_{α_i} or S_{α_i} with α_i exceeding the following threshold in order to discourage the opponent's moves altogether:

$$\alpha_i^{\mathsf{threshold}} = \max\left(\frac{1}{2k_{-i}}, \frac{k_{-i} + c_{-i} - \sqrt{c_{-i}(2k_{-i} + c_{-i})}}{k_{-i}^2}\right).$$

While this is desirable, it is sometimes infeasible to play fast enough if the state checking or moving costs are high. A different preventive approach for a player is to somehow make it increasingly more difficult for its opponent to take over the resource over time. When the level of difficulty reaches some threshold, its opponent will au-

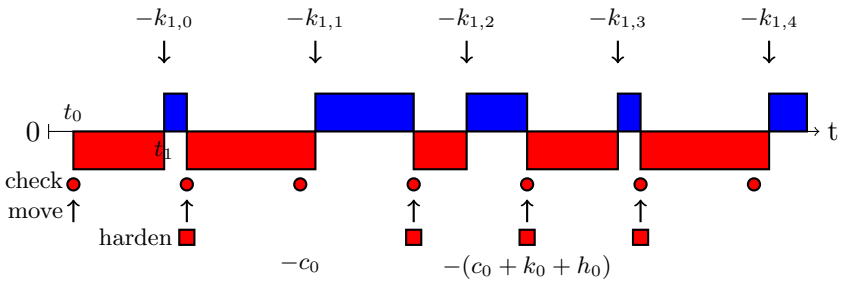


Figure 2.3: A FlipIt(H, P) game with the attacker's flipping cost over time $k_{1,i} < k_{1,i+1}$.

tomatically cease playing, and thus resulting in a long-term benefit for the player. In FlipIt type of games, this can be modelled by having a player spending an additional *periodic hardening cost* h_i every time it regains control, so that the opponent would have to spend more and more whenever trying to take over the resource. In particular, we define a new class of strategies, called *periodic hardening*, denoted by

$$H = \{H_{\alpha,h} | \alpha, h > 0\}$$

where H_{α_i,h_i} represents player i's strategy in which it performs state checking at rate α_i and every time it regains control, player i also spends a cost of h_i to make subsequent take-overs by the opponent more difficult/costly (Figure 2.3). This cost h_i could feature, for example, some penetration testing process that results in vulnerabilities being patched, similar to that modelled in [23]. It modifies the net utility of player i who performs state checking with hardening as follows, with $m_i(t)$ being the number of state checks occurred prior to t:

$$B_i(t) = G_i(t) - (k_i + h_i)n_i(t) - c_i m_i(t).$$

Our focus in the section is to study how the defender may efficiently spend the budget for control hardening in response to the attacker's strategy. Because of the hardening process which aims at stopping the attacker's moves/flips at some point in time, players' expected utilities do not remain the same over time, and thus the use of average benefit as players' preferences over outcomes does not accurately reflect reality. It is rather suggestive that we represent such preferences via either players' net benefit or net loss over a finite time interval (long enough to practically capture "infinity"). The following result establishes the net loss of the defender, which entails the cost of all security operations and the loss of control over the resource (to the attacker). We inherently use this as the defender's "utility" in order to capture its preferences over the set of hardening strategies H.

Lemma 2.2. *Consider the game* $\mathsf{FlipIt}(H, P)$[4]. *Let* $(H_{\alpha_0,h_0}, P_{\alpha_1}) \in H \times P$ *be an action profile, and* s *be the number of effective*[5] *moves/flips that the attacker makes following* $(H_{\alpha_0,h_0}, P_{\alpha_1})$. *The defender's expected net loss at time* $t \geq (s+1)\max(\frac{1}{\alpha_0}, \frac{1}{\alpha_1})$ *is:*

$$L_0(t) = G(\alpha_0, \alpha_1, t) + \begin{cases} s\left[\dfrac{1}{2\alpha_0} + k_0 + h_0\right] & \text{if } \alpha_0 \geq \alpha_1, \text{ or} \\ s\left[\left(1 - \dfrac{\alpha_0}{2\alpha_1}\right)\dfrac{1}{\alpha_0} + k_0 + h_0\right] + z(s, \alpha_0, \alpha_1) & \text{otherwise,} \end{cases}$$

for some function G *independent of* s, *and* z *such that for all* $s, s' \in \mathbb{N}^+$ *it holds that* $|z(s) - z(s')| \leq \frac{\alpha_0}{6\alpha_1^2}$.

Proof. We first notice that the defender's net loss comes from several factors: the loss of control over the resource, the cost of state checking, the cost of moving, and the cost of hardening. We compute such loss in two cases:

- $\alpha_0 \geq \alpha_1$: this also implies $\delta_0 = 1/\alpha_0 \leq 1/\alpha_1 = \delta_1$. Let $t_1 \in [0, \delta_1]$ be the first moment when the attacker makes a move/state-check. We use the same argument as in (2.2) to show that the defender's expected control time within $[t_1 + i\delta_1, t_1 + (i+1)\delta_1]$ is $\delta_1(1 - r/2)$ for $i \in \{0, \ldots, s-1\}$ and $r = \alpha_1/\alpha_0$. This means that the defender's expected loss in control time over the period $[t_1, t_1 + s\delta_1]$ is

$$L_0^{control} = s\left(\delta_1 - \delta_1\left(1 - \frac{r}{2}\right)\right) = s\left(\delta_1 - \delta_1\left(1 - \frac{\alpha_1}{2\alpha_0}\right)\right) = \frac{s}{2\alpha_0}$$

  On the other hand, the defender's expected number of state-checks over the same period is $L_0^{check} = s\delta_1\alpha_0 c = s\frac{\alpha_0}{\alpha_1}c$. Because each period of the form $[t_1 + i\delta_1, t_1 + (i+1)\delta_1]$ is preceded by a state-check (and move if necessary) by the attacker, such period starts by the attacker's control over the resource, i.e., $C_1(t_1 + i\delta_1) = 1$. Also, since $\delta_0 \leq \delta_1$, it is guaranteed that within such period the defender will certainly perform a state-check, and subsequently a move/flip. Despite (possibly) many state-checks, only one move would be made by the defender within this period. Therefore, the total number of defender's moves/flips within $[t_1, t_1 + s\delta_1]$ is s, and thus its loss due to moving/flipping is $L_0^{flip} = k_0 s$. Finally, the same argument yields the loss due to hardening, which is $L_0^{harden} = h_0 s$. The defender's

[4]Since our main focus is on the defender's strategies, in this section we omit the attacker's state-checking strategies since S_{α_1} and P_{α_1} bring identical effect to the defender's utility.

[5]By effective move we mean a move that results in a change of control from one player to another.

net loss within $[t_1, t_1 + s\delta_1]$ is

$$L_0^{control} + L_0^{check} + L_0^{flip} + L_0^{harden} = s\left[\frac{1}{2\alpha_0} + k_0 + h_0 + c_0\frac{\alpha_0}{\alpha_1}\right] \tag{2.10}$$

Taking the expected value of the above over the choice of t_1 yields the same value as it does not depend on t_1. For the defender's loss over other time periods, i.e., $[0, t_1)$ and $(t_1 + s\delta_1, t)$, we notice that they end and start with the defender's control of the resource, respectively. In other words, $C_0(t') = 0$ for $t' < t_1$ or $t' > t_1 + s\delta_1$. Also, since the attacker makes no move within these periods, the defender's expected loss is only in the cost of state checking, i.e.,

$$g(t_1, \alpha_0, \alpha_1, t) = c_0(t_1 + t - s\delta_1)\alpha_0 = c_0(t_1 + t - \frac{s}{\alpha_1})\alpha_0$$

Due to uniformly random phase selection in the attacker's first move/state-check, the defender's expected net loss before the attacker's first move and after its last move is :

$$\int_0^{\delta_1} \frac{1}{\delta_1} g(t_1, \alpha_0, \alpha_1, t)dt = c_0 t\alpha_0 + \frac{c_0\alpha_0}{2\alpha_1} - \frac{c_0 s\alpha_0}{\alpha_1} = G(\alpha_0, \alpha_1, t) - \frac{c_0 s\alpha_0}{\alpha_1}.$$

Summing the above with (2.10) we get the lemma statement for $\alpha_0 \geq \alpha_1$.

- $\alpha_0 < \alpha_1$: this also implies $\delta_0 = 1/\alpha_0 > 1/\alpha_1 = \delta_1$. We first notice that because $t \geq (s+1)\max(\delta_0, \delta_1)$, the number of defender's moves is exactly s in response to s flips by the attacker. Hence the expected loss due to flipping, state-checking, and hardening are the same as before, i.e.,

$$k_0 s + t c_0 \alpha_0 + h_0 s \tag{2.11}$$

Let t_0 be the first moment the defender performs a state-check. We notice that within $[0, t_0]$ the attacker moves/flips at most once because the defender's first move is at t_0, meaning that there are at least $s-1$ attacker's moves after t_0. Therefore, using the same argument as in (2.2) we may infer that the defender's expected loss in control time between its two consecutive state-check $[t_0 + i\delta_0, t_0 + (i+1)\delta_0]$ for $i \in [0, s-2]$ is $\delta_0(1 - \frac{\alpha_0}{2\alpha_1})$. The defender's expected loss in control time within $[t_0, t_0 + (s-1)\delta_0]$ is thus

$$(s-1)\delta_0\left(1 - \frac{\alpha_0}{2\alpha_1}\right) = \frac{s-1}{\alpha_0}\left(1 - \frac{\alpha_0}{2\alpha_1}\right). \tag{2.12}$$

Next we consider the defender's expected loss within the interval $[t_0+(s-1)\delta_0, t_0+s\delta_0]$. Unlike previous intervals of the same length, an attacker's move does not occur in this interval with certainty. Indeed, if the attacker makes a move before t_0, then it has already finished s moves before reaching this interval. We thus only consider the case in which the attacker does not move before t_0, which also means that $t_0 \in [0, \delta_1]$ and that the attacker's first move is $t_1 \in [t_0, \delta_1]$. This allows us to capture the loss of control in $[t_0+(s-1)\delta_0, t_0+s\delta_0]$ by the following:

$$z(s, \alpha_0, \alpha_1) = \int_0^{\delta_1} \frac{1}{\delta_0} \int_{t_0}^{\delta_1} \frac{1}{\delta_1} \left(s\delta_0 + t_0 - \left(t_1 + \left\lceil \frac{(s-1)\delta_0 + t_0 - t_1}{\delta_1} \right\rceil \delta_1 \right) \right) dt_1 dt_0 \tag{2.13}$$

where $t_1 + \left\lceil \frac{(s-1)\delta_0+t_0-t_1}{\delta_1} \right\rceil \delta_1$ is the first attacker's flip/state check within $[t_0 + (s-1)\delta_0, t_0 + s\delta_0]$. It is not difficult to see that $z(s, \alpha_0, \alpha_1)$ is bounded by the following expressions, to which their difference is:

$$\frac{1}{\delta_0\delta_1} \int_0^{\delta_1} \int_{t_0}^{\delta_1} (\delta_0 + t_0 - t_1)\, dt_1 dt_0 - \frac{1}{\delta_0\delta_1} \int_0^{\delta_1} \int_{t_0}^{\delta_1} (\delta_0 - t_1)\, dt_1 dt_0 = \frac{\delta_1^2}{6\delta_0} = \frac{\alpha_0}{6\alpha_1^2}$$

The remaining interval to be considered is $[0, t_0]$. It is easy to see that the defender's expected loss within this interval is independent of s, and hence we denote it as $L_0^{control}(t_0) > 0$. Based on (2.11) and (2.12) define G as

$$G(\alpha_0, \alpha_1, t) = L_0^{control}(t_0) - \frac{1}{\alpha_0} \left(1 - \frac{\alpha_0}{2\alpha_1} \right) + tc_0\alpha_0$$

and we thus succeed in constructing G and z that satisfy the lemma for $\alpha_0 < \alpha_1$.

□

In the next step we model the correlation between the hardening cost and the number of attacks that would eventually happen. Such correlation captures the effectiveness of the hardening process, and is vital to measuring the optimal hardening cost in any situation. Particularly, the value of s is influenced by the attacker's benefit from a move, the original move cost, and the hardening cost. We model the relation among these variable by a a *cost update function* f, such that at the i-th attack (attacker's move), the attack cost becomes $f(k_1, h_0, i-1)$. Attacker's moves/flips stop at the $(s+1)$-th attempt if the cost involved is greater than the attacker's expected control, which is expressed below.

Lemma 2.3. *Consider the game* $\mathsf{FlipIt}(H, P)$. *Let* H_{α_0,h_0} *and* P_{α_1} *be strategies of the defender and attacker, respectively. Let* f *be a* cost update function such *that the*

attacker's move cost after i moves is $f(k_1, h_0, i)$. Suppose that the attacker stops at the first attack of which the cost is greater than the expected gain, then the attacker's actual number of moves is minimum s such that :

$$f(k_1, h_0, s) \geq \begin{cases} \delta_0/2 & \text{if } \alpha_0 \geq \alpha_1, \text{ or} \\ \delta_0 - \delta_1/2 - z(s, \alpha_0, \alpha_1) + z(s+1, \alpha_0, \alpha_1) & \text{otherwise,} \end{cases} \tag{2.14}$$

where z is same as in Lemma 2.2.

Proof. The proof of this lemma follows the same line as that of Lemma 2.2. Consider the case $\alpha_0 \geq \alpha_1$. The attacker's $(s+1)$-th attack/move/flip is followed by the period $[t_1 + s\delta_1, t_1 + (s+1)\delta_1]$, where t_1 is the time of the attacker's first move. Because $\delta_0 = 1/\alpha_0 < 1/\alpha_1 = \delta_1$ and due to uniformly random phase, the attacker's expected control time would be $\delta_0/2$ based on (2.2). This proves to the lemma statement for $\alpha_0 \geq \alpha_1$. For the case $\alpha_0 < \alpha_1$ we notice from the proof of Lemma 2.2 that within the period $[t_0 + (s-1)\delta_0, t_0 + s\delta_0]$ the attacker's expected control time is $z(s, \alpha_0, \alpha_1)$. However without limit on attacks it should normally be $\delta_0 - \delta_1/2$ following (2.2). This means that $\delta_0 - \delta_1/2 - z(s, \alpha_0, \alpha_1)$ is the attacker's expected control time resulted from the $(s+1)$-th attack within the same time interval. However, the $(s+1)$-th attack may also occur in the next interval, i.e., $[t_0 + s\delta_0, t_0 + (s+1)\delta_0]$, which yields $z(s+1, \alpha_0, \alpha_1)$ as the attacker's expected control time. Summing up the two expressions give the lemma statement for $\alpha_0 < \alpha_1$. □

In reality, the structure of f strongly depends on how control of the resource can be hardened. For demonstration, we consider two distinct examples of constructions for f as below:

$$f_1(k_1, h_0, i) = k_1 + i\lambda h_0 \quad \text{and} \quad f_2(k_1, h_0, i) = k_1 + i\frac{\mu h_0}{h_0 + \lambda}. \tag{2.15}$$

The former construction captures a linear relation between the hardening cost and the attack cost, with $\lambda \geq 0$ being the effectiveness of the hardening process. In the context of information security, this happens, for instance when the resource contains a large number of identical but also independent subsystems, so that the control becomes more secure as more subsystems are hardened. In that case, the attacker would only control the resource if it manages to compromise most (if not all) of these subsystems. A real-world example of such resource is a multi-party system used for secret-sharing [118]. The attacker might only get the secret if it compromises most of parties involved in the secret-sharing scheme. The latter construction f_2 follows an idea similar to that from Gordon and Loeb [63], in which the cost of attack is also raised ($f_2'(h_0) > 0$),

but at a decreasing rate ($f_2''(h_0) < 0$). In other words, it becomes increasingly difficult to raise attack cost as security approaches optimality. The continuous decrease in the increasing rate implies an upperbound on the maximum achievable attack cost. This is facilitated by $\mu \geq 0$, the maximum increase in attack cost after each attack, whereas $\lambda > 0$ represents the effectiveness of the hardening process. Construction f_2 also agrees with [24] and [23] which suggest a weakest-link model in which attack cost increases linearly to the number of steps. This means that optimisation of security is more effective if it is done over many steps/attacks because the defender receives information that would be useful for the hardening process: the exploited vulnerabilities. We express this property of f_2 in the lemma below, in which the former statement indicates that knowledge about exploited vulnerabilities makes the process of hardening easier, whereas the latter signifies that such knowledge also contributes directly to increasing the attack cost.

Lemma 2.4. *Let* $f_2(k_1, h_0, i) = k_1 + i\frac{\mu h_0}{h_0+\lambda}$, *then for all* $k_1, h, h', \mu, \lambda \in \mathbb{R}^+$ *and* $i, i' \in \mathbb{N}^+$ *then the following hold:*

$$f_2(k_1, h, i) = f_2(k_1, h', i') \wedge i > i' \Longrightarrow i'\frac{\partial f_2}{\partial h_0}(h, i) > i\frac{\partial f_2}{\partial h_0}(h', i') \wedge h < h' \tag{2.16}$$

$$h \cdot i = h' \cdot i' \wedge i > i' \Longrightarrow f_2(k_1, h, i) > f_2(k_1, h', i') \tag{2.17}$$

Proof. We prove the first statement by showing that $h < h'$. Indeed,

$$f_2(k_1, h, i) = f_2(k_1, h', i') \wedge i > i' \Rightarrow i\frac{\mu h}{h+\lambda} = i'\frac{\mu h'}{h'+\lambda} \wedge i > i' \Rightarrow \frac{\mu h}{h+\lambda} < \frac{\mu h'}{h'+\lambda}$$

However, $\frac{\mu h_0}{h_0+\lambda}$ is increasing in h_0 since its derivative is $\frac{\lambda\mu}{(h_0+\lambda)^2} > 0$, and hence $h < h'$. The other part of the first statement follows straightforwardly:

$$i'\frac{\partial f_2}{\partial h_0}(h, i) = i'i\frac{\lambda\mu}{(h+\lambda)^2} > i'i\frac{\lambda\mu}{(h'+\lambda)^2} = i\frac{\partial f_2}{\partial h_0}(h', i')$$

The second statement is also straightforward, as $hi = h'i'$ and $i > i'$ implies $h < h'$, therefore

$$f_2(k_1, h, i) = k_1 + i\frac{\mu h}{h+\lambda} = k_1 + i'\frac{\mu h'}{h+\lambda} > k_1 + i'\frac{\mu h'}{h'+\lambda} = f_2(k_1, h', i').$$

□

Next we study the optimal hardening strategy of the defender given its chosen rate of assessment α_0 as well as that other attacker, i.e., α_1. We give two results respectively for our two construction of cost update function f_1 and f_2.

Proposition 2.3. *Consider the game* $\mathsf{FlipIt}(H, P)$ *with cost update function* $f_1(k_1, h_0, i) = k_1 + i\lambda h_0$. *Let* α_0 *and* α_1 *be the rates of chosen by the defender and the attacker, respectively. The defender's optimal choice of* h_0 *is such that the attacker stops after at most one attack.*

Proof. Let $B(s, \alpha_0, \alpha_1) := \mathsf{RHS}((2.14))$ denote the right-hand side of (2.14). In this case $B(s, \alpha_0, \alpha_1)$ is the attacker's expected gain as the result of the s-th attack/move. Suppose that s is the number of attacker's moves, then

$$f_1(k_1, h_1, s) \geq B \iff k_1 + s\lambda h_0 \geq B \iff h_0 \geq \frac{B - k_1}{\lambda s}$$

From the expression of the defender's expected net loss as in Lemma 2.2 it is easy to see that the defender should select $h_0 = \frac{B-k_1}{\lambda s}$ as the optimal hardening cost given s. Consider a different number of attacker's moves $s' > s$ with corresponding optimal hardening cost $h_0' = \frac{B-k_1}{\lambda s'}$. Consider the case $\alpha_0 \geq \alpha_1$, we have

$$\frac{\partial L(t)}{\partial s} = \frac{\partial \left(G(\alpha_0, \alpha_1, t) + s \left[\frac{1}{2\alpha_0} + k + \frac{B-k_1}{\lambda s} \right] \right)}{\partial s} = \frac{1}{2\alpha_0} + k_0 > 0$$

Therefore it is optimal that $s = 1$ and $h_0 = \frac{B-k_1}{\lambda}$. For the case $\alpha_0 < \alpha_1$ we have

$$\begin{aligned}
L_0(t, s') - L_0(t, s) &= (s' - s) \left[\frac{2\alpha_1 - \alpha_0}{\alpha_1 \alpha_0} + k_0 \right] + h_0' s' - h_0 s + z(s', \alpha_0, \alpha_1) - z(s, \alpha_0, \alpha_1) \\
&= (s' - s) \left[\frac{2\alpha_1 - \alpha_0}{\alpha_1 \alpha_0} + k_0 \right] + z(s', \alpha_0, \alpha_1) - z(s, \alpha_0, \alpha_1) \\
&\geq (s' - s) \frac{2\alpha_0 - \alpha_0}{\alpha_1^2} + z(s', \alpha_0, \alpha_1) - z(s, \alpha_0, \alpha_1) \\
&\geq (s' - s) \frac{\alpha_0}{6\alpha_1^2} + z(s', \alpha_0, \alpha_1) - z(s, \alpha_0, \alpha_1) \\
&\geq \frac{\alpha_0}{6\alpha_1^2} - \frac{\alpha_0}{6\alpha_1^2} = 0
\end{aligned}$$

This again concludes that it is optimal for the defender to pick h_0 such that $s = 1$, in which case $h_0 = \frac{B-k_1}{\lambda}$. □

Proposition 2.4. *Consider the game* $\mathsf{FlipIt}(H, P)$ *with cost update function* $f_2(k_1, h_0, i) = k_1 + i\frac{\mu h_0}{h_0 + \lambda}$. *Let* α_0 *and* α_1 *be the rates chosen by the defender and the attacker, respectively. Define the following:*

$$n_a = \frac{\mathsf{RHS}((2.14)) - k_1}{\mu} \quad \textit{and} \quad L^* = k_0 + \frac{1}{\alpha_0} \max\left(\frac{1}{2}, 1 - \frac{\alpha_0}{2\alpha_1} \right).$$

Assume that $z(s, \alpha_0, \alpha_1)$ is identical[6] *for all $s \in \mathbb{N}^+$, the following hold:*

- *the number of attacker's moves is at least $\lceil n_a \rceil$.*
- *the optimal hardening cost is*

$$h_0 = \frac{n_a \lambda}{s - n_a} \text{ where } s = \left\lceil n_a + \frac{1}{2}\left(\frac{\sqrt{L^* + 4n_a^2\lambda}}{\sqrt{L^*}} - 1\right)\right\rceil \tag{2.18}$$

is the corresponding number of attacks.

Proof. Let $B(s, \alpha_0, \alpha_1) := \mathsf{RHS}((2.14))$ we formulate the number of attacks s:

$$f_2(k_1, h_0, s) \geq B \iff k_1 + s\frac{\mu h_0}{h_0 + \lambda} \geq B \implies s = \left\lceil \frac{(B - k_1)(h_0 + \lambda)}{\mu h_0} \right\rceil$$

Note that similar to the proof of Proposition 2.3, it is optimal to choose h_0 such that the above satisfies and that $s \in \mathbb{N}^+$. The minimum number of attacker's moves is:

$$\lim_{h_0 \to \infty} s(h_0) = \lim_{h_0 \to \infty} \left\lceil \frac{(\lambda + h_0)(B - k_1)}{\mu h_0} \right\rceil = \left\lceil \frac{B - k_1}{\mu} \right\rceil = \lceil n_a \rceil$$

For the second statement of the proposition, we notice that

$$L_0(h_0, t) = G(\alpha_0, \alpha_1, t) + (h_0 + L^*) \left\lceil \frac{(\lambda + h_0) n_a}{h_0} \right\rceil + \begin{cases} 0 & \text{if } \alpha_0 \geq 0, \text{ or} \\ z(s, \alpha_0, \alpha_1) & \text{otherwise.} \end{cases} \tag{2.19}$$

which is minimised when $h_0 = (\lambda + h_0) n_a / m$, or equivalently $h_0 = \frac{\lambda n_a}{m - n_a}$ for some positive integer $m \geq \lceil n_a \rceil$ due to the fact that $l'(h_0) < 0$ for $l(h_0) = \frac{(\lambda + h_0) n_a}{h_0}$. In this case, m is the number of the attacker's moves. Minimisation of $L_0(h_0, t)$ over $h_0 \in \mathbb{R}^+$ is equivalent to minimisation of the following over $m \in \mathbb{N}^+$:

$$\mathbb{L}_0(m) = (h_0(m) + L^*)m = m\left(\frac{\lambda n_a}{m - n_a} + L^*\right).$$

Since $\mathbb{L}_0'(m) = L^* - \frac{n_a^2 \lambda}{(m - n_a)^2}$ with $m \in \mathbb{R}$ has at most two roots, $\mathbb{L}_0(m)$ has at most one globally minimum point. A solution $m^* \in \mathbb{R}^+$ of the equation $\mathbb{L}_0(m) = \mathbb{L}_0(m + 1)$ would (if exists) allows minimisation of $\mathbb{L}_0(m)$ at $m = \lceil m^* \rceil \in \mathbb{N}^+$. We thus proceed

[6] We assume this in order to simplify the result and proof. Later on in the discussion we will reconcile the result with the fact that $z(s, \alpha_0, \alpha_1)$ is not identical for all $s \in \mathbb{N}^+$, which in fact will point out interesting insights.

[6] This loss includes the attacker's occupation of the resource and the cost spent on protecting the resource.

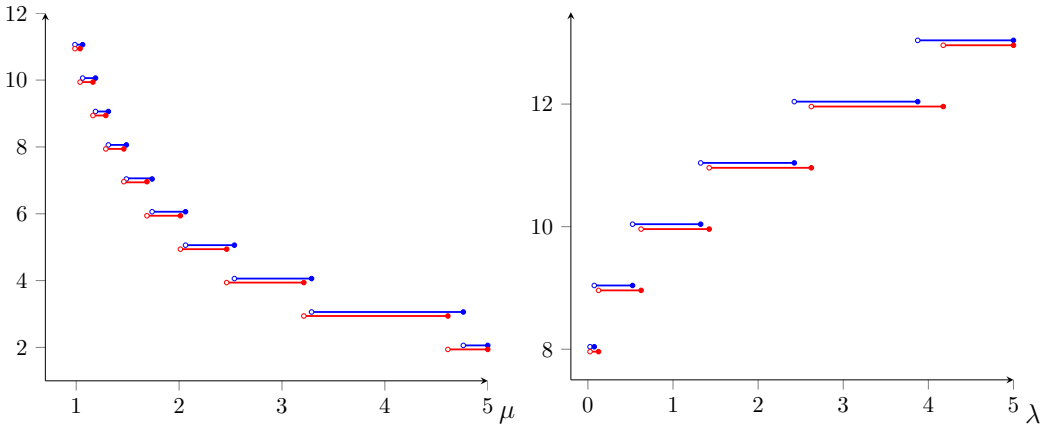
to compute m in the following

$$\begin{aligned}
\mathbb{L}_0(m^*) = c_0(m^*+1) &\iff m^*\left(\frac{\lambda n_a}{m^*-n_a}+L^*\right) = (m^*+1)\left(\frac{\lambda n_a}{m^*+1-n_a}+L^*\right) \\
&\iff \frac{n_a^2\lambda}{(m^*-n_a)(m^*+1-n_a)} - L^* = 0 \\
&\iff m^* = n_a + \frac{1}{2}\left(\pm\frac{\sqrt{L^*+4n_a^2\lambda}}{\sqrt{L^*}} - 1\right) \\
&\implies m = \lceil m^* \rceil = \left\lceil n_a + \frac{1}{2}\left(\frac{\sqrt{L^*+4n_a^2\lambda}}{\sqrt{L^*}} - 1\right)\right\rceil
\end{aligned}$$

which completes the proof of Proposition 2.4. Note that we only take the greater solution of m^* because $\lim_{m\to\pm\infty}\mathbb{L}_0(m) = \pm\infty$ implies that the greater solution of m^* is the minimum point whereas the other is the maximum point. □

Discussions. The above propositions stress a need for appropriate decision over the investment for hardening the resource control. In information security, hardening may mean, for example, system patching, penetration testing, adding security layers, etc. Proposition 2.3 suggests that when the attack cost can be raised linearly to the hardening cost, then it is best for the defender to spend enough in improving security (after a breach) of the resource once and for all, so that attacks no longer occurs. In the provided example, this means hardening all involved subsystems even if only some of them were breached in the previous incident. This is due to the fact that the effectiveness of hardening remains constant over time, and thus the sooner it is done the better. On the other hand, Proposition 2.4 provides different insights into the effects of non-linear cost update function f_2 on hardening strategies. First of all, Proposition 2.4 assumes the homogeneity of $z(s, \alpha_0, \alpha_1)$ over $s \in \mathbb{N}^+$. While the expression of z is given in (2.13), it essentially captures the potential advantage of the defender in terms of "better-than-uniform" knowledge about the attacker's phase, which was chosen at uniformly random before the game. Indeed, if the defender's phase is $t_0 \in [0, \delta_1]$, and if a state check at t_0 does not reveal an attack, then the defender can infer that the attacker's phase is $t_1 \in [t_0, \delta_1]$ rather than $t_1 \in [0, \delta_1]$. By allowing homogeneity of $z(s, \alpha_0, \alpha_1)$ over s, Proposition 2.4 provides optimal hardening strategies whilst ignoring the advantage of "better-than-uniform" knowledge. It consequently allows us to emphasise the effect of such knowledge to hardening strategies, which we demonstrate in Figure 2.4. This figure points out three important insights:

- By increasing the bound μ, the defender tends to perform more aggressive hardening and aims at stopping sooner. One may think of μ as an indicator of the



(a) (b)

Figure 2.4: Example number of attacks under optimal hardening strategies taking into account (red plot) and ignoring (blue plot) $z(s, \alpha_0, \alpha_1)$, given parameters $\alpha_0 = \frac{1}{15}, \alpha_1 = \frac{1}{14.5}, k_0 = 2, k_1 = 0.1$, with (a) $\lambda = 0.3$ and $\mu \in [1, 5]$, or (b) $\mu = 1$ and $\lambda \in [0, 5]$.

potential of the hardening process, in which case the figure agrees with the fact that the defender should "squeeze" such potential as much as possible to stop attacks as soon as possible.

- By raising the parameter λ, the defender becomes more relax and tends to prolong the hardening process, thus suffering from more attacks. As an explanation, we notice that λ has significant contribution to the effectiveness of hardening at low hardening cost h_0. This can be thought of as "boost" in security improvement at the beginning of every step, and starts to fade as more is invested. Such a boost is mainly due to discovery of a vulnerability following an exploit/breach, which in this case is the attacker's move.

- The knowledge about attacker's phase as represented by $z(s, \alpha_0, \alpha_1)$ has considerable effects on the optimal hardening strategies, which in this case results in the number of attacks being reduced in many cases.

2.7 Conclusion

In this chapter we investigate the concern on the choices of long-term strategic security plans for protecting organisational assets. These choices are represented by questions such as "are we vulnerable?" and "are we compromised?" This concern has become increasingly more important for large businesses as well as governmental units in the era where attackers are advanced, and have the resources to be persistent. To do so,

we extend the FlipIt game between an attacker and a defender periodically taking over a resource from each other, with the trade-off between the cost of taking over, and the duration of the control. In our model, in addition from taking over, we allow players to check who is controlling the resource. We compare between blind take-over strategies and those that involve "check first, then take over", and show a threshold for the checking cost, under which the latter tactic is preferred.

In further extensions, we study strategic plans on how organisations would rationally invest in security improvement to discourage attackers. Our analysis on specific models proposed suggests that there are cases in which a system must suffer from many attacks to become sufficiently secure to deter attackers. In reality, this is because security breaches serve as valuable information for improving system security. In another aspect, we relax our hidden assumption so that state checking might be incorrect, and study not just the frequency of security assessment, but also how quality-price-ratio may even discourage assessment of security. Since our models mostly deal with the defender's utility, the lessons learned may apply to not just advanced persistent threats (APTs), but also a pool of non-persistent threats that occurs with known frequency, e.g., from a community of underground hackers.

Chapter 3

Strategic Information Sharing in Competitive Environments

In this chapter, we study incentives behind investments by competing companies in discovery of their security vulnerabilities and sharing of their findings. Specifically, we consider a game between competing firms that utilise a common platform in their systems, where the goal is to unveil and fix vulnerabilities of this platform. The game consists of two stages: firms must decide how much to invest in researching vulnerabilities, and thereafter, how much of their findings to share with their competitors. We fully characterise the Perfect Bayesian Equilibria (PBE) of this game, and translate them into realistic insights about firms' strategies. Further, we develop a monetary-free sharing mechanism that encourages both investment and sharing, a missing feature when sharing is arbitrary or opportunistic. This is achieved via a light-handed mediator: it receives a set of discovered bugs from each firm and moderate the sharing in a way that eliminates firms' concerns on losing competitive advantages. This research provides an understanding of the origins of inefficiency and paves the path towards more efficient sharing of cyber-intelligence among private companies.

3.1 Introduction

Businesses across different sectors of the economy, from telecommunication and finance to energy, healthcare and transportation, increasingly rely on cyberspace and IT services. Past incidents of cyber-attacks and consequent damages have left little doubt in the minds of business managers and policy makers about the importance of investment in cyber-security. Gathering and exchange of security intelligence is identified as a key factor in enhancing the effectiveness of individual cyber-security measures.

Steps have been taken by governments to provide the environments to galvanise and coordinate exchange of cyber-security information across private companies: UK launched the "cyber-security Information Sharing Partnership" [117] after a pilot program in 2011/12 as a "joint, collaborative initiative between industry and government to share cyber threat and vulnerability information in order to increase overall situational awareness of the cyber threat". In the US, the "National Coordinating Center for Communications (NCC)" acts as the "Information Sharing and Analysis Center (ISAC)" for telecommunications [108].

While "Information Sharing and Analysis Centers (ISACs)" – such as Information Technology (IT)-ISAC and Financial Services (FS)-ISAC – can provide the platform for exchange of cyber-intelligence, the role of incentives cannot be ignored. Providing the means of communication in the presence of strategic and competing profit-maximising entities does not necessarily lead to exchange of their cyber-security information. In order to understand the incentives of firms in creating and sharing information security knowledge, it is important to identify the distinct nature of the security information being shared. Some example categories of the type of cyber-intelligence to be shared are: (a) steps, protocols and measures a firm has taken to improve its security; (b) past incidents of successful or unsuccessful attacks and the resulting privacy, intellectual property and financial losses; and (c) discovered security vulnerabilities. Sharing each of these types of information have specific incentive implications. For instance, "public disclosure" of security breach incidents can harm the consumers and investors' confidence and lead to a statistically significant decreases in the market value of firms [29,34,60]. In this work, we particularly focus on the third type of information: sharing discovered security vulnerabilities, or *bugs* for short.

From the societal point of view, sharing knowledge of security vulnerabilities among firms is a positive move: it improves the overall efficiency of discovery efforts of the vulnerabilities. It moreover enhances the cyber protection of an entire industry against future cyber-attacks by reducing the common exploitable threats. It is often the case that different organisations of an economic sector bear similar vulnerabilities and face similar threats in their information systems [90]. This is partly due to the adoption of common implementations, libraries or operating systems across different organisation. For instance, the `Heartbleed` bug (formally, CVE-2014-0160), a buffer-over-read vulnerability in the `OpenSSL` cryptographic library exposed in April 2014, affects around half a million certificates issued by trusted certificate authorities [102]. Another reason why different technological companies face common threats is the incorporation of discovered vulnerabilities into hacking toolkits which enables even less sophisticated users to configure the same malware to attack across different organisations [90].

Recognising the need for cyber-protection, different companies invest in finding their security vulnerabilities. These can be "bugs" for example in their application level software, operating system or implementation of a network protocol, which we will generally refer to as the common *platform*. No company knows exactly how many bugs there are in a software they are using. More investment and effort in security research increases the chances of discovering them, but there is always a factor of luck involved. Each company patches and rectifies the vulnerabilities it finds, which is usually the much easier part than finding them in the first place. Each bug that is not discovered by a company, and hence not rectified, is potentially exploitable by cyber-attackers.

When a bug is indeed successfully exploited, the victim suffers direct losses. These can include outage of their service, recovery costs, losses of important data, user compensation, legal fines, etc. However, a company may also be affected by incidents of cyber-attacks on other companies in that economic sector: On one hand, the whole sector of economy may suffer a blow: as customers may lose confidence in the whole "service" and seek alternative "safer" means. For instance, if one or a few major online banking companies are hit by a cyber-attack, then some customers may lose confidence in the whole sector and switch to traditional banking altogether. Moreover, investors and stock holders may too lose confidence in the whole industry in favour of alternative options for investment. These two effects translate to a net market value loss of the whole sector, which bites all of the companies upon a successful attack on anyone of them. However, on the other hand, if (and once) a bug is exploited in competitor(s) that a company has discovered before (and has hence taken care of), it can have the opposite effect of boosting the confidence of customers as well as the investors: customers may switch to use and investors redirect their capital to the "safer" company. In other words, discovering a bug in a common software may give a company a "competitive edge" compared to others.

The two effects work in the opposite direction of each other in terms of incentives for sharing the found vulnerabilities. The sharing strategies, in turn, affect the investment decisions to discover the bugs in the first place: On the one hand, sharing information translates to a more effective discovery process and hence encourages investment, as the findings of one company is fortified by another's since the process of finding the bugs is probabilistic in nature. But on the other hand, there can be a tendency of free-riding on the discovery investment of other companies and hence get away with less investment. Further complicating the problem is the presence of uncertainty and information asymmetry: companies ought to make their discovery investment decisions in the face of uncertainties about the total number of bugs, and they need to make

decision about sharing of their findings not knowing the number of findings of the other company.

This chapter is organised as follows: In Section 3.3, we model the interdependent security research investment and information sharing decisions of two strategic and competing firms as a two stage Bayesian game. We fully determine the Perfect Bayesian Equilibria of the game in closed-form in Section 3.4. Specifically, in Subsection 3.4.1, we derive the Bayesian equilibrium strategies of the firms about sharing of their finding for a given investment pair, and given their findings. In particular, we establish that sharing strategies are unique and dominant strategies in the simple forms of "full-sharing" or "no sharing", completely determined by the competitive nature of the security findings. In Subsection 3.4.2, we derive the investment strategies of the firms knowing their subsequent sharing strategies. We show how "full sharing" leads to free-riding and inefficiently low investments. Also how "no sharing" is socially inefficient by preventing mutual benefit of sharing, double-efforts and potential over-investment. Finally, in Section 3.5, we provide a light-weight mediation mechanism free of monetary-transfers that enable (partial) sharing of the information when the firms fail to achieve any sharing on their own.

3.2 Related Work

Information sharing in general has appeared very early in modern economic research, particularly on trade associations (e.g. in [52, 136, 143]) where the effect of information sharing is captured as improvement in the efficiency of production, i.e., reducing the marginal cost of production, or improving demand, or both. On the other hand, affects of security and security breaches on organisations also receive extensive attention from academics with different focus, for example, privacy [36], data integrity [8], password security [154], and secure applications [139]. In a seminal work, Gordon and Loeb [63] present a model for optimising security investment, taking into account potential loss due to security breaches on different vulnerabilities.

Information sharing in the context of cyber-security is investigated in several notable works. As an example, Gordon et al. [64] consider a game in which two firms must decide how much to invest in security and how much information to share. Their work examine incentives to information sharing, as well as how it affects both expenditures in security and the overall level of security in presence of freeriding alongside cross-firm cooperation. In an extension, Gal-Or and Ghose [53] study the effects of the same problem, however on competing firms trying to sell their products. Here they add to the firms' strategies a choice of product price after investment of security and

information sharing. With such model their work produce several implications on how information sharing and security investment affect demands and price competition on firms. Another recent work is by Liu et al. [89] with a focus on designing an information sharing network such as FS-ISAC that encourage firms to participate, using mechanisms such as member fees and insurance plans. Other works also exist, e.g., by Xiong and Chen [147] for repeated games, Hausken [71], and Gao et al. [54] on games between the social planner, firms and the attacker.

Our contributions. As a common feature among aforementioned models, there is no specification of the type of security information to be shared. The decision of how much information to share is modelled as a normalised continuous variable between zero and one, zero corresponding to no sharing and one corresponding to full sharing. Also, these works consider investment on perimeters/mechanisms that harden the security of firms' information infrastructures.

In contrast, we consider the question of how much firms should invest security research, i.e., gaining information about the security of their platforms. An example of such investment is by Google, especially on its Project Zero [50]. The reason why firms (especially large ones) should make such investment is rather obvious. As outstanding targets for zero-day exploits [97], large firms should actively uncover their vulnerabilities instead of waiting for the community which largely involve potential attackers. Nevertheless, this type of investment is more connected to information sharing, as such research results can be shared among firms that utilise the same platforms (e.g., Linux OS, Apache web server). Indeed, the relation between security investments and information sharing is rather loose in the previous literature. For instance, the effective amount of shared information is heuristically chosen as the product of the investment decision and sharing decision [64].

In order to meaningfully capture sharing of information and connection investment, we do not model information by the [0,1] scale as in other works, but as the discovered security vulnerabilities by each player, and hence, the sharing decisions in our model is the "number" of bugs to be shared. Our work specifically models the relation between the investment strategies for "generation" of security information (via security research) and that of sharing them. Moreover, we develop a mediation mechanism that enables sharing in the face of competition as a novel contribution.

3.3 Model

Our model considers a game between firm i and firm j, denoted by $N = \{i, j\}$, where each decides how much to invest in security research on a common "platform", and subsequently how many of their found security vulnerabilities to share with the other. The platform has an unknown number of security vulnerabilities, or "bugs", which, if not discovered and rectified, may be exploited with ramifications for both firms. Before the game starts, the nature determines the total number of bugs following some distribution. Let the random variable representing the total number of bugs be B with the sample space of $\mathbb{N}^{+1}$ and known mean value λ. The realisation of B is not observed by any of the firms. The game play consists of two stages: *investment* and *sharing*, as described in the following:

1. **Investment**: In this stage, the players, while unaware of the total number of bugs in the platform, "simultaneously" decide how much to invest in bug discovery, and make it publicly known. Note that simultaneous move in the context of game theory just implies that neither one of the players can assume pre-commitment to a decision by the other players. A player's investment c determines the probability $p \in [0, 1)$ that each bug is discovered. For simplicity, we assume that the bugs are homogeneous, in that they are equally difficult to discover. Moreover, we assume discovery of each bug is independent across the bugs and across the firms. The research investment c and discovery probability p are related through function π as $p = \pi(c)$, with $\lim_{c\to\infty} \pi(c) = 1$[2]. We naturally assume that $\partial\pi(c)/\partial c > 0$, as well as $\partial^2\pi(c)/\partial c^2 \leq 0$: The chance of finding bugs should be improved with more investment, and it is increasingly more difficult to improve the success of bug discoveries. In general we assume that the two firms have distinct cost-probability relations, denoted as $\pi_i(c)$ and $\pi_j(c)$. Because we assume both π_i and π_j are strictly increasing, there is a one-to-one mapping between investment and discovery probability. Indeed, $c_i = \pi_i^{-1}(p_i)$ and $c_j = \pi_j^{-1}(p_j)$. Hence, we can equivalently represent each player's strategy in this stage by its choice of discovery probability, i.e., p_i and p_j.

2. **Sharing**: After investments are made, each player privately and independently "discovers" some bugs in the platform. Also both players are informed of each

[1] We adopt the convention that random variables are denoted by capital letters and their realisations by lower case. Also, $\mathbb{N}^+ := \mathbb{N} \cup \{0\}$.

[2] This constraint captures a famous consensus: there is no such thing as perfect security [129]. Indeed, the fact that $\pi(c) = 1$ would imply that all security vulnerabilities are discovered, which should not be possible in reality. Hence we let this happen at the cost c at ∞.

others' investment decisions [3]. Subsequently, each independently decides how many of its findings to share with the other. Note that the discoveries are not part of the strategies of the players and is rather determined probabilistically –by "nature"– once the investments are made. Since the discoveries are private, they cause an "incompleteness of information" of players about each other. From game-theoretic viewpoint this sharing stage appears naturally as a Bayesian game. In particular, firms i and j respectively discover N_i and N_j bugs in the platform, which are random variables with the common sample space of $\{0, 1, \ldots, B\}$.[4] The set of discovered bugs may have an overlap, i.e., some identical bugs may be discovered by both firms. We denote the number of common bugs by N_{ij}. The sample space of N_{ij} is $\{0, 1, \ldots, \min(N_i, N_j)\}$. Given the total number of bugs B and investment levels c_i and c_j, the nature determines the number of bugs discovered by each firm and the number of commonly discovered bugs N_i, N_j and N_{ij}. The quadruple (B, N_i, N_j, N_{ij}) is the random variable over the set of possible "states of the world" Ω. Note that due to the revelation of investments p_i and p_j at the end of the first stage, the probability distribution $\mathcal{D}(\lambda, p_i, p_j)$ of (B, N_i, N_j, N_{ij}) over Ω is publicly known. For each nature state $(b, n_i, n_j, n_{ij}) \in \Omega$, firm i (resp. j) observes n_i (resp. n_j), i.e., the number of bugs it has discovered, as its "type". For each realisation of the number of found bugs and announced investments, a firm must decide how many of its found bugs to share with the other. Due to the homogeneity assumption of bugs, the bugs to be shared can be assumed to be picked uniformly randomly. A (pure) strategy of firm i is thus a mapping $s_i(p_j, n_i) : [0, 1] \times \mathbb{N}^+ \to \mathbb{N}^+$ such that $s_i(p_j, n_i) \leq n_i$.[5] Let $\sigma_i = (p_i, s_i)$ denote the pure strategies of player i for *the whole game*. After both σ_i and σ_j are decided, the overall utilities of each player is determined as the result of its investment together with the expected losses/gains from security incidents.

In what follows, we describe the expected utility of the two players after two stages of actions. We assume risk-neutral players, that is, the players care equally about their utility of expected outcome and their expected utility. Hence, the utilities are linear sums of the (negative of the) expected costs per each bug minus the investment cost for discovery of the bugs. Note that at the time of taking the decision about sharing the discovered bugs, the investments for discovering the bugs are "sunk" costs, i.e., they are already spent and will not affect the cost to go of different actions to take. Each bug, if not discovered by or informed to a player, will be successfully exploited on

[3] This feature of our model also reflects reality in which organisations inform the public about their investment in security research, e.g., Microsoft [13], IBM [73].

[4] By $\{0, 1, \ldots, B\}$, it is meant that given the realisation $B = b$, the set is $\{0, 1, \ldots, b\}$.

[5] Since p_i is part of player i's strategy, it needs not be included as an argument to s_i.

that player by attackers with a probability, which without loss of generality, we take to be one. We assume that the exploitation probabilities and the severity of bugs are homogeneously distributed. For each bug there are three types of losses/damages:[6]

- **Direct loss** $\boxed{l > 0}$: affecting only the compromised firm (e.g. outage/denial of its services, compromise/corruption of its data, etc.).
- **Market shrinkage** $\boxed{\tau \geq 0}$: the common loss as a result of a successful attack that affects both, even the firm that is not compromised. This is the effect of the market shrinkage after a successful attack as a result of a portion of both demand and investment moving away from (abandoning) the whole service/technology in favour of "safer" alternatives, or simply relinquishing that sector altogether.
- **Competitive loss** $\boxed{\delta \geq 0}$: when *only one* firm is compromised by attackers, the compromised firm loses δ while the other gains δ. This represents the shifting of demand and/or public investment (stocks) upon a successful attack.

Given the notions described above, there are four possibilities of net cost for each bug that a player may incur: (a) The bug is known by both players (either through own discovery or through the information shared by the other firm). In this case, the utility of the players is $(0, 0)$, as neither one of the players loses anything.[7] (b) The bug is known by player i, but not player j. In this case, the utility pair is $(\delta - \tau, -\delta - \tau - l)$: the bug will be exploited at firm j, which causes its direct loss l and a competitive advantage δ for firm i, while both of them will lose τ due to market shrinkage. (c) The bug is known by player j, but not player i. This is the mirror situation to case-b: the utility pair is $(-\delta - \tau - l, \delta - \tau)$. (d) The bug is known by neither player. Here, there is no competitive advantage of one over the other, but there is still the market shrinkage effect, besides the direct losses to both. Hence, the utilities are $(-\tau - l, -\tau - l)$.

To facilitate the computation of the expected utilities, we define the following auxiliary random variables (as also depicted by a Venn diagram in Figure 3.1): let $B_{i,j}$, $B_{i,\neg j}$, $B_{\neg i,j}$ and $B_{\neg i,\neg j}$ represent the number of bugs that, respectively, both players, only player i, only only player j, and neither player knows about. Let the (expected) utility of players be denoted by u, which is a function from the strategy profile of the players and the state of the world to the set of real numbers. The expectation is taken with respect to the realisation of $B_{i,j}$, $B_{i,\neg j}$, $B_{\neg i,j}$ and $B_{\neg i,\neg j}$ given B, N_i, N_j and N_{ij}, and the sharing strategies. We are now ready to compute the expected utility of

[6] For simplicity of exposition, we assume the losses and damages are symmetrical; it is straightforward to generalise the results to non-symmetric cases.

[7] The assumption is that once the bug is discovered, its "fix" is immediate and costless.

Table 3.1: List of main notations

Parameter	Definition
B, b	Random variable for the total number of bugs, and a realisation
N_i, n_i	Random variable for the number of bugs discovered by i, and a realisation
N_{ij}	Random variable for the number of common bugs discovered by both
a_i	Action of player i: how many discovered bugs to share
λ	Expected number of the total number of bugs
p_i, p_j	Probability that each bug is discovered by player i,j
u_i, u_j	Expected utilities of player i, j
c_i, c_j	Discovery investment cost of player i,j
l	Direct loss upon exploitation of an (undiscovered) bug by attackers
δ	Loss (gain) in utility of the player who is the only one attacked (not attacked) – capturing the market competition effect
τ	Loss in utility of both players if a bug is exploited in either one of them – capturing the total market section shrinkage effect
$p = \pi(c)$	The relation relating the level of investment c to the discovery probability of a bug p. In this work, we use $p = \pi(c) = 1 - e^{-\theta c}$.

player i given a realisation of the state of the world $\omega = (b, n_i, n_j, n_{ij})$, and $\sigma_i = (p_i, s_i)$, $\sigma_j = (p_j, s_j)$:

$$u_i(\omega, \sigma_i, \sigma_j) = -c_i(p_i) + 0 \cdot B_{i,j} + (\delta - \tau) \cdot B_{i,\neg j} + (-\delta - \tau - l) \cdot B_{\neg i,j} + (-\tau - l) \cdot B_{\neg i,\neg j} \quad (3.1)$$

In what follows we analyse further the structure of this utility function and derive the "outcome" of the game and study its properties. For readers' reference, we summarise our game notion below.

Definition 3.1. *A vulnerability sharing game is a game played between players in the set $N = \{i, j\}$ consisting of two stages:*

1. *Investment: i and j independently select $p_i, p_j \in [0, 1)$, respectively.*
2. *Sharing: players are informed with p_i, p_j and participate in a Bayesian game $\langle N, \Omega, \langle A_i, T_i, C_i, \tau_i, \mathfrak{p}_i, u_i \rangle_{i \in N} \rangle$ with $N = \{i, j\}$, $\Omega = \{(b, n_i, n_j, n_{ij}) \mid b, n_i, n_j, n_{ij} \in \mathbb{N}^+ \wedge n_i, n_j \leq b \wedge n_{i,j} \leq \min(n_i, n_j)\}$, $A_i = A_j = \mathbb{N}^+$, $T_i = T_j = \mathbb{N}^+$, $C_i = C_j = C$:*

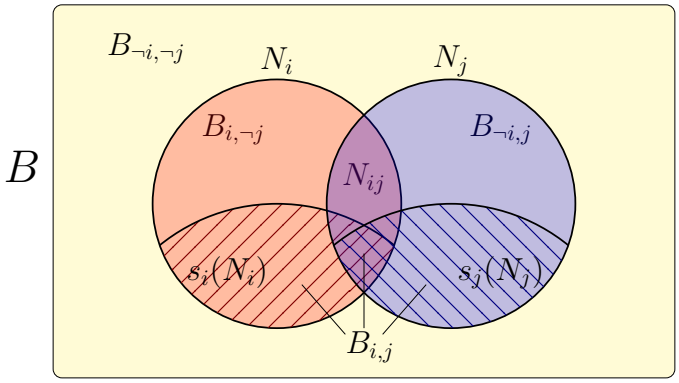


Figure 3.1: Venn diagram illustration of the sets of bugs.

$\mathbb{N}^+ \to 2^{\mathbb{N}^+}$ *such that* $C(t) = \{0\} \cup [t]$, $\tau_i : \Omega \to T_i$ *such that* $\tau_i(b, n_i, n_j, n_{ij}) = n_i$ *and similarly* $\tau_j(b, n_i, n_j, n_{ij}) = n_j$, $\mathfrak{p}_i = \mathfrak{p}_j$ *following a probability distribution* $\mathcal{D}(\lambda, p_i, p_j)$ *over* Ω, *and* u_i *as in* (3.1).

3.4 Analysis of the Game

When dealing with strategic entities with inter-dependent utilities, investigating equilibria, most notably Nash Equilibria, is a method of predicting their decisions. Our game contains sequential moves, and thus an ordinary Nash equilibrium concept would potentially cause the problem of "non-credible threats". Also note that our game contains simultaneous actions in each stage, and hence is of "imperfect information". We therefore examine possible perfect Bayesian equilibria (PBE), a solution concept that effectively eliminates non-credible threats in sequential games with incomplete and imperfect information.

In short, a PBE is a strategy profile such that, given any player's belief about the game history that is consistent with that profile, then that player's remaining part of strategy (from the belief onward) is its best response. To find the set of PBEs, we notice from Definition 3.1 that since the investment decisions are announced before sharing, each Bayesian game in the second stage is a proper subgame of the whole game. This means that we can use backward induction and first construct (s_i, s_j) as a Bayesian Nash equilibrium (BNE) of the Bayesian game in the second stage for all choices of p_i and p_j. This in turn determines the utility of the players for each choice of (p_i, p_j), which allows us to build a simple strategic-form game with actions p_i and p_j corresponding to the first stage of the game. The remaining task will be to find a Nash equilibrium for this game, which will lead to a proper PBE for the whole two-stage game. We thus proceed by studying the second stage of the game

(information sharing), and then proceed to analyse players' investments given their equilibrium sharing strategies.

3.4.1 Second Stage: Sharing the Bug Discoveries

The first step of the analysis is to construct the expected values of players' utilities (3.1) from the probability distribution $\mathcal{D}(\lambda, p_i, p_j)$ over the states of the Nature. This can effectively be done by computing the expected values of $B_{i,j}$, $B_{i,\neg j}$, $B_{\neg i,j}$ and $B_{\neg i,\neg j}$. Since $\mathbb{E}(B_{i,j})$ is multiplied by zero, we can safely ignore it. For the rest, we have:

$$\mathbb{E}[B_{i,\neg j}|\omega, \sigma_i, \sigma_j] = (n_i - n_{ij})(1 - \frac{s_i(p_j, n_i)}{n_i}) \tag{3.2a}$$

$$\mathbb{E}[B_{\neg i,j}|\omega, \sigma_i, \sigma_j] = (n_j - n_{ij})(1 - \frac{s_j(p_i, n_j)}{n_j}) \tag{3.2b}$$

$$\mathbb{E}[B_{\neg i,\neg j}|\omega, \sigma_i, \sigma_j] = b - n_i - n_j + n_{ij} \tag{3.2c}$$

In (3.2a),(3.2b), we have in part used the fact that the bugs to be shared are chosen uniformly randomly across the discovered bugs. Replacing in (3.1), we obtain:

$$\begin{aligned}\mathbb{E}\left[u_i(\omega, \sigma_i, \sigma_j)\right] = -c_i(p_i) + (\delta - \tau)(n_i - n_{ij})(1 - \frac{s_i(p_j, n_i)}{n_i}) + \\ (-\delta - \tau - l)(n_j - n_{ij})(1 - \frac{s_j(p_i, n_j)}{n_j}) + (-\tau - l)(b - n_i - n_j + n_{ij})\end{aligned} \tag{3.3}$$

We are looking for strategy profiles (strategy pairs (s_i, s_j) in our two-player context) that are simultaneous best responses to each other, given the information that each player has, notably including its number of discovered bugs. In the Bayesian Nash equilibria of the game, each candidate strategy for a player must be a maximiser of its expected utility given the strategy of the other player and given its observed type (number of discovered bugs).[8] Formally, for a given p_i and p_j, we are looking for the strategy pairs (s_i^*, s_j^*), such that:

$$\forall n_i \in \mathbb{N}^+,\ s_i^*(p_j, n_i) \in \arg\max_{s_i(p_j, n_i)} \mathbb{E}[u_i(\omega, (p_i, s_i(p_j, n_i)), (p_j, s_j^*(p_i, n_j)))|n_i] \tag{3.4}$$

and simultaneously vice versa for j. Such pairs constitute the (pure) Bayesian Nash Equilibria of the second stage of our game. The pair (s_i^*, s_j^*) is further, a Dominant

[8] To analyse the game, each player must specify its actions for all of its possible types, and not just the realised (and observed) type. This is because, the expected utility of each player depends on the possible actions of the other player(s) weighted against their potential types, since the type of other player(s) are not directly observed.

(pure) Bayesian Nash Equilibrium iff:

$$\forall n_i \in \mathbb{N}^+, \forall s_j,\ s_i^*(p_j, n_i) \in \arg\max_{s_i(p_j,n_i)} \mathbb{E}[u_i(\omega, (p_i, s_i(p_j, n_i)), (p_j, s_j(p_i, n_j)))|n_i] \tag{3.5}$$

and simultaneously vice versa for j. We are now ready to express the main result of this section:

Proposition 3.1. *Suppose $p_i, p_j < 1$. If $\delta < \tau$, the unique dominant pure Bayesian Nash Equilibrium of the second stage of the game is $(s_i^*(p_j, n_i), s_j^*(p_i, n_j)) = (n_i, n_j)$, i.e., sharing all the discovered bugs. If $\delta > \tau$, it is $(s_i^*(p_j, n_i), s_j^*(p_i, n_j)) = (0, 0)$, i.e., sharing no information at all. When $\delta = \tau$, any strategy pair becomes a Bayesian Nash Equilibrium. This proposition holds irrespective of the distribution of the total number of bugs.*

Proof. According to (3.5), a pair (s_i^*, s_j^*) constitutes a Dominant Bayesian Equilibrium if, for each type of a player, its corresponding action is the best (provided the knowledge of its type), irrespective of the strategy of the other player. From (3.3), the only term in the expression of $u_i(\omega, \sigma_i, \sigma_j)$ that involves s_i is the second term: $(\delta - \tau)[(n_i - n_{ij})(1 - s_i(p_j, n_i)/n_i)]$. Hence, with the assumption of $p_j < 1$ in mind, the maximisation of $\mathbb{E}[u_i(\omega, \sigma_i, \sigma_j)|n_i]$ with respect to $s_i(p_j, n_i)$ reduces to maximising $(\delta - \tau)(1 - s_i(p_j, n_i))$, which yields the proposition.[9] □

Discussion. The proposition makes intuitive sense: when $\delta > \tau$, each bug that is only known by a player wins it a strictly positive (expected) competitive gain of $(\delta - \tau)$, as the competitive shift in the demand and public investment outweighs the overall drop in the demand and fall in the stock market of the whole market section. Hence it rather not share any of its findings, irrespective of what the other player chooses. This is because the players have no means of making their decisions "contingent" on the decision of the other.[10] Similarly, when $\delta < \tau$, the competitive shift in the demand and capital, falls short of the whole market section shrinkage. Therefore, the players prefer to share all their findings to (selfishly) keep themselves from being hurt. Perhaps the surprising result is that the dominant strategy of the players turned out to be

[9] Although the proposition leaves out the cases in which the condition $p_i, p_j < 1$ are not satisfied, they are not difficult to analyse: suppose $p_j = 1$, then $\mathbb{E}[(n_i - n_{ij})(1 - s_i(p_j, n_i)/n_i)|n_i] = 0$, and hence the expression for $\mathbb{E}[u_i(\omega, \sigma_i, \sigma_j)|n_i]$ will not depend on s_i at all. Hence, in any Bayesian Nash Equilibria, the choice of s_i becomes arbitrary. Similar situation happens for s_j when $p_i = 1$. Intuitively, if the other player "knows every bug for certain", then a player cannot affect its utility through its action: it cannot gain any competitive advantage if $\delta > \tau$, or help prevent market shrinkage when $\delta > \tau$. Note that realistically, we can safely assume $p_i, p_j < 1$, as no practical amount of investment leads to absolute certainly of finding all bugs.

[10] We will see in §3.5 how this situation can be altered in the presence of a mediator.

completely determined by the relative values of only two parameters δ and τ. This proposition fully determines the sharing strategy of the firms. Notably, aside from the special case of $\delta = \tau$, the equilibrium is unique and hence, there is no ambiguity in selection of the equilibrium.

Apparently, our result only serve as relaxed prediction of firms' behaviour in reality. This is mainly due to the simplicity of our model along with its assumptions. For example, if losses (l, δ, and τ) associated with different vulnerabilities are not identical, the conditions in Proposition 3.1 might need to be stricter. Likewise, changes to this result might be observed if losses between two firms are not symmetric as in our model.

Next, we investigate how each firm invests for discovering the bugs knowing the subsequent sharing strategies.

3.4.2 First Stage: Investment for Bug Discovery

In the first stage of the game, each player decides about its investment amount for the discovery of bugs, heeding the strategy of the other player in the second stage. To obtain closed-form results, we need to model the relation between investment decision and the chance of finding bugs. A simple candidate for such relation is the following: $p = \pi(c) = 1 - e^{-\theta c}$, where θ represents a measure of the efficiency of the investment: a larger θ corresponds to a higher efficiency of the investment. As the level of investment increases to infinity, the probability of discovery of each bug asymptotically approaches unity. The two firms may be different in how "efficient" they are in their investment. A firm with more prepared talents can expect higher chances of discovery with less investment. To capture the potential heterogeneity in the investment efficiencies, we consider two potentially different θ_i and θ_j. Our investment-discovery probability relation has the extra property that the relative efficiency of the investment stays constant for all investment values, specifically: $(\partial\pi_i/\partial c)/(\partial\pi_j/\partial c) = \theta_i/\theta_j$. This relation can also be equivalently represented in its inverse form: $c_i(p_i) = -\ln(1-p_i)/\theta_i$ for $p_i \in [0, 1)$, and likewise for j. Note that the condition of Proposition 3.1 $p_i, p_j < 1$ is automatically satisfied when $\lim_{p\to 1} c(p) \to \infty$, as is the case in our example.

To analyse this stage, we note that Proposition (3.1) fully determines (s_i^*, s_j^*) for each profile of (p_i, p_j). This allows us to treat the first stage as a "one-shot" game of investment with action profiles of the form (p_i, p_j).

3.4.3 The Case of $\delta < \tau$

For the case of $\delta < \tau$, from Proposition 3.1, the dominant strategy of both players is to share *all* of their findings, i.e., $s_i(p_j, n_i) = n_i$ and $s_j(p_i, n_j) = n_j$ for all $n_i, n_i \in \mathbb{N}^+$.

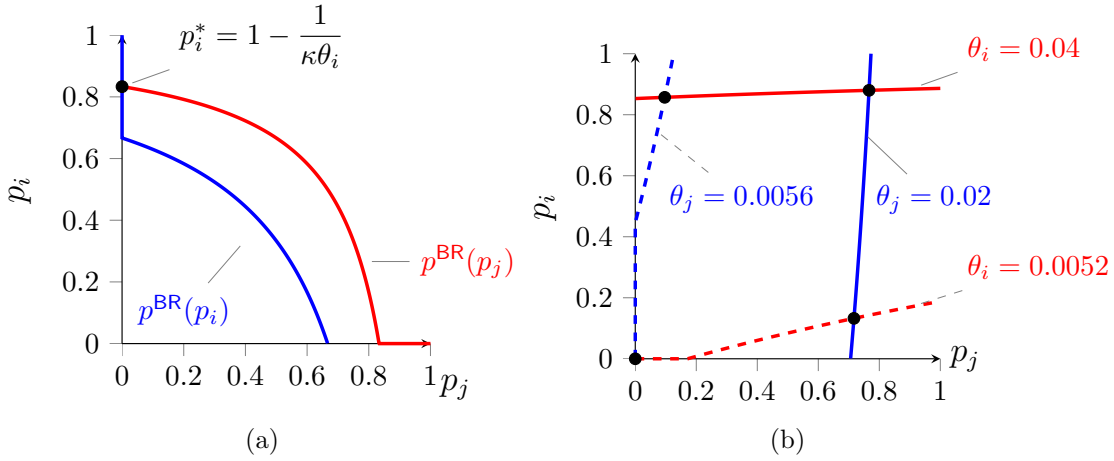


Figure 3.2: (a) Example best response curves for the case of $\delta < \tau$, investigated in §3.4.3. In the figure $\theta_i > \theta_j$. The intersection gives the simultaneous best response pair in the first stage of the game as: $(p_i^*, p_j^*) = ([1 - (\kappa\theta_i)^{-1}]^+, 0)$. The parameters used are: $\lambda = 100$, $\tau = 0.5$, $l = 1$, $\theta_i = 0.04$, $\theta_j = 0.02$. (b) Example best response curves for the case of $\delta < \tau$ and different θ_is and θ_js.

Then, the second and third terms in (3.3) become zero, and we get:

$$\begin{aligned}\mathbb{E}[u_i(\omega, (p_i, s_i^*), (p_j, s_j^*))] &= -c_i(p_i) + (-\tau - l)\mathbb{E}[B - N_i - N_j + N_{ij}] \\ &= -c_i(p_i) + (-\tau - l)\lambda(1 - p_j)(1 - p_i)\end{aligned}$$

The best response p_i^{BR} as a relation over p_j is hence:

$$p_i^{\mathsf{BR}}(p_j) = [c_i'^{-1}(\kappa(1 - p_j))]^+, \text{[11]} \qquad \text{where } \kappa := \lambda(\tau + l). \tag{3.6}$$

Note that when $p_i^{\mathsf{BR}} > 0$, $\partial p_i^{\mathsf{BR}}/\partial p_j = -\kappa/c_i''(p_i^{\mathsf{BR}}) < 0$, i.e., more investment by the other player leaves *less* incentive for a player to invest. Similarly, we have: $\mathbb{E}[u_i(\omega, (p_i, s_i^*), (p_j, s_j^*))] = -c_j(p_j) + (-\tau - l)(1 - p_i)\lambda(1 - p_j)$, and hence: $p_j^{\mathsf{BR}}(p_i) = [c_j'^{-1}(\kappa(1 - p_i))]^+$. The fixed points of the best response correspondence $(p_i, p_j) \rightrightarrows ([c_i'^{-1}(\kappa(1 - p_j))]^+, [c_j'^{-1}(\kappa(1 - p_i))]^+)$ constitute the outcome of the first stage. For our example cost function $c(p) = -\ln(1 - p)/\theta$, the simultaneous best response in (3.6)translates to the following (Figure 3.2a):

$$p_i^{\mathsf{BR}}(p_j) = [1 - \frac{1}{\theta_i\kappa(1 - p_j)}]^+, \qquad p_j^{\mathsf{BR}}(p_i) = [1 - \frac{1}{\theta_j\kappa(1 - p_i)}]^+.$$

[11] We use the conventions: $f'(x) := df(x)/\,dx$ and $a^+ := \max\{0, a\}$.

This, together with Proposition 3.1, lead to the following result:[12]

Proposition 3.2. *If $\delta < \tau$ and $\theta_i > \theta_j$, the Perfect Bayesian Equilibrium (PBE) of the two-stage game is $((p_i^*, s_i^*(p_j, n_i)), (p_j^*, s_j^*(p_i, n_j))) = (([1 - \frac{1}{\kappa\theta_i}]^+, n_i), (0, n_j))$ for all $n_i, n_j \in \mathbb{N}^+$ and all $p_i, p_j \in [0, 1)$, where $\kappa := \lambda(\tau + l)$. That is, only the more efficient firm invests in discovery of the bugs to achieve discovery probability of $[1 - (\kappa\theta_i)^{-1}]^+$ – and all the findings are then shared.*[13]

Discussion. The less efficient firm free-rides on the bug discovery investment of the more efficient company, knowing that all the findings will be shared. This might leap the reader to the conclusion that the PBE outcome is socially inefficient simply because of the existence of "free-riding". However, a social planner may also prefer that the investment is done by the more efficient firm as opposed to distributing the investment among both, hence garnering a higher social return on the aggregate investments. In what follows, we will evaluate the social utility and the socially efficient outcome and compare the two.

Investigating social welfare:

Let W represent the expected (utilitarian) *social utility*, defined simply as the sum of the expected utilities of the two firms, i.e., $W := u_i + u_j$.[14] First off, it is straightforward to argue that in the socially optimal outcome, all the findings are shared (the social utility can only be improved by sharing the findings, as the investment decisions are now disentangled from the sharing decisions). The expected optimal social utility is hence as follows:

$$\mathbb{E}[W] = -c_i - c_j - 2(\tau + l)\mathbb{E}[B_{\neg i, \neg j}] = -c_i(p_i) - c_j(p_j) - 2\kappa(1 - p_i)(1 - p_j) \qquad (3.7)$$

In our sample cost function, maximising $\mathbb{E}[W]$ hence yields: $(\hat{p}_i, \hat{p}_j) = ([1-(2\kappa\theta_i)^{-1}]^+, 0)$. Comparing the socially optimal solution with the PBE outcome, we have $\hat{p}_j = p_j^* = 0$, and when $2\kappa\theta_i > 1$, we have: $\hat{p}_i > p_i^*$. That is, to maximise the social utility (sum of the expected utilities of the two firms), the less efficient firm, as in the PBE outcome, makes no investment free-rides on the investment of the more efficient firm. However,

[12] The exact values of the investments depend on the cost function adopted, however, the qualitative observations hold for a wide class of such functions.

[13] When $\theta_i = \theta_j = \theta$, i.e., the two firms are homogeneous in terms of their efficiencies of bug discovery investments, the equilibrium point is not unique and becomes the set: $\{(p_i^*, p_j^*) \in [0, 1]^2, p_i^* = [1 - (\theta\kappa(1 - p_j))^{-1}]^+\}$.

[14] Other notions of social welfare exist, e.g., the egalitarian objective $W := \min(u_i, u_j)$.

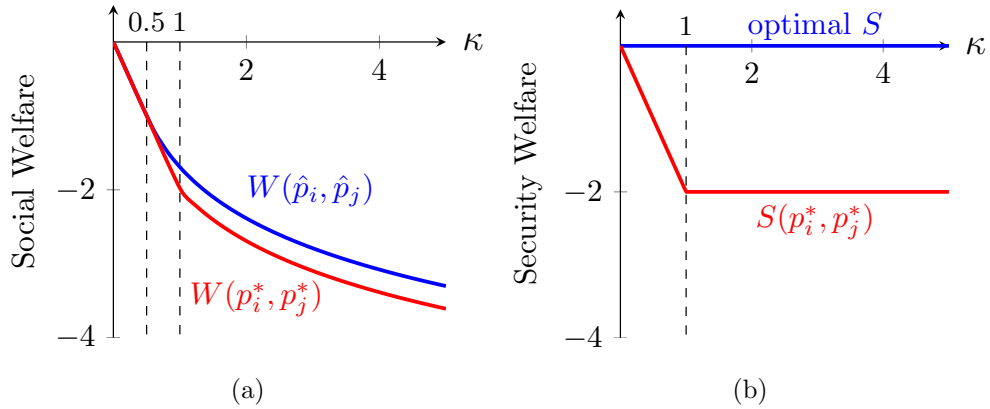


Figure 3.3: Example depiction of the optimal and achieved social welfare (3.3a) and security utility (3.3b) for the case of $\delta < \tau$ as functions of $\kappa = \lambda(\tau + l)$.

compared to the PBE outcome, the more efficient firm invests more. This makes intuitive sense: the less efficient firm offers a lower return on investment (offers less "return" in turning investment into probability of bug discovery) and hence should not invest at all. Instead, the investments must be made by the more efficient firm and all the findings be shared. Moreover, the more efficient firm must consider the aggregate losses and invest more carrying the burden of the two, compared to the PBE, where it only considers the effect of its investment on its own losses. Note that even when the players are homogeneous in terms of their efficiencies, i.e., when $\theta_i = \theta_j$, the socially optimal investment turns out to choose only one of the firms to invest. This is because if both firms independently research, they may waste efforts if the results turn out to be the same, e.g., they discover the same bugs. The value of the optimum social welfare is:

$$W(\hat{p}_i, \hat{p}_j) = -\ln(2\kappa\theta_i)/\theta_i - 1/\theta_i \text{ for } \kappa\theta_i > 1/2, \text{ and: } -2\kappa \text{ for } \kappa\theta_i \leq 1/2. \tag{3.8}$$

The social welfare that is achieved at the equilibrium outcome of the game is:

$$W(p_i^*, p_j^*) := -\ln(\kappa\theta_i)/\theta_i - 2/\theta_i \text{ for } \kappa\theta_i > 1, \text{ and: } -2\kappa \text{ for } \kappa\theta_i \leq 1. \tag{3.9}$$

An example comparison between the two is depicted in Figure 3.3a.

Here, we define another metric of social welfare in the context of economics of network security. Let the *security utility* u^S of a player be the negative of the costs of security attacks. Security utility, such defined, is related to the utility of a player as $u^S = u + c$, that is, it includes all the security damages but excludes the investment cost. Now, let the *security welfare* S, as a metric of the aggregate security of the two firms, be the sum of their security utilities: $S := u_i^S + u_j^S$. The security utility is

related to the utilitarian social welfare in the following way: $S = W + c_i(p_i) + c_j(p_j)$. The optimal S is achieved by picking $p_i = 1$ and sharing all the findings, which yields $S = 0$. Figure 3.3b illustrates a comparison between the achieved security utility at the equilibrium and the optimal S.

Comparative statics

[15] Recall from Proposition (3.2), that for $\delta < \tau$, in part we have: $(p_i^*, p_j^*) = ([1 - 1/(\kappa\theta_i)]^+, 0)$. Hence, as long as $\delta < \tau$, $\theta_i > \theta_j$ and $p_i^* > 0$ (i.e., for $1 < \kappa\theta_i$), we have the following straightforward observations:

$$\frac{\partial p_i^*}{\partial \tau}, \frac{\partial p_i^*}{\partial l}, \frac{\partial p_i^*}{\partial \lambda}, \frac{\partial p_i^*}{\partial \theta_i} > 0, \qquad \frac{\partial p_j^*}{\partial \tau}, \frac{\partial p_j^*}{\partial l}, \frac{\partial p_j^*}{\partial \lambda}, \frac{\partial p_j^*}{\partial \theta_j} = 0.$$

We also have $\partial p_i^*/\partial \theta_j = 0$, and perhaps most interesting of all $\partial p_i^*/\partial \delta = 0$; intuitively, player i shares all of its findings and thus removes any dependence of its utility (and hence its best strategy) on δ. Also, note that even though $\partial p_i^*/\partial \theta_i > 0$, i.e., more efficiency in investment means higher choice of probability of discovery, this does not necessarily translate to higher choice of investment. In fact, we have: $\partial c_i(p_i^*)/\partial \theta_i < 0$ for $1 < \kappa\theta_i < e$, and $\partial c_i(p_i^*)/\partial \theta_i > 0$ for $\kappa\theta_i > e$. Moreover, from (3.9), for $p_i^* > 0$ we have: $W^* := W(p_i^*, p_j^*) = -\ln(\kappa\theta_i)/\theta_i - 2/\theta_i$ and $S^* := S(p_i^*, p_j^*) = -2/\theta_i$. Hence, when $\delta < \tau$, $\theta_i > \theta_j$ and $1 < \kappa\theta_i$, we have:

$$\frac{\partial W^*}{\partial \tau}, \frac{\partial W^*}{\partial l}, \frac{\partial W^*}{\partial \lambda} < 0, \ \frac{\partial W^*}{\partial \theta_i} > 0, \qquad \frac{\partial S^*}{\partial \tau}, \frac{\partial S^*}{\partial l}, \frac{\partial S^*}{\partial \lambda} = 0, \ \frac{\partial S^*}{\partial \theta_i} > 0.$$

3.4.4 The Case of $\delta > \tau$

Following Proposition 3.1, the dominant strategy of the players in the second stage is to share *none* of their findings, i.e., $s_i(p_j, n_i) = 0$ and $s_j(p_i, n_j) = 0$ for all $n_i, n_i \in \mathbb{N}^+$ and all $p_i, p_j \in [0, 1)$. Then from (3.3), we obtain:

$$\begin{aligned}\mathbb{E}[u_i(\omega, (p_i, s_i^*), (p_j, s_j^*))] = &-c_i(p_i) + (\delta - \tau)\lambda p_i(1 - p_j) \\ &+ (-\delta - \tau - l)p_j\lambda(1 - p_i) + (-\tau - l)(1 - p_j)\lambda(1 - p_i) \quad (3.10)\end{aligned}$$

The best response relation for player i is therefore:

$$p_i^{\mathsf{BR}}(p_j) = [c_i'^{-1}(\lambda(\delta + l + p_j\tau))]^+.$$

[15] In economics, *comparative statics* is the study of the change in the "equilibrium" outcome when a change in a parameter is/would be introduced.

A point to observe is that for $p_i^{\mathsf{BR}} > 0$, we have: $\partial p_i^{\mathsf{BR}}/\partial p_j = \lambda\tau/c_i''(p_i^{\mathsf{BR}}) > 0$, i.e., more investment by the other player leads to *more* investment by a player. This is in sharp contrast to the previous case of $\delta < \tau$. Similarly: $p_j^{\mathsf{BR}}(p_i) = [c_j'^{-1}(\lambda(\delta + l + p_i\tau))]^+$. For our example cost function, the simultaneous best response is therefore the solution the following system (Figure 3.2b):

$$p_i^{\mathsf{BR}}(p_j) = [1 - \frac{1}{\theta_i\lambda(\delta + l + p_j\tau)}]^+, \qquad p_j^{\mathsf{BR}}(p_i) = [1 - \frac{1}{\theta_j\lambda(\delta + l + p_i\tau)}]^+. \tag{3.11}$$

Straightforward algebraic investigation reveals that the solution is unique and given as follows:

$$\text{If } \Delta \geq 0\text{: } \begin{cases} p_i^* = \dfrac{\left[-\lambda\theta_i\theta_j((\delta+l)^2 - \tau^2) + \tau(\theta_i - \theta_j) + \sqrt{\Delta}\right]^+}{2\tau\theta_i\theta_j(\delta + l + \tau)} \\ p_j^* = \dfrac{\left[-\lambda\theta_i\theta_j((\delta+l)^2 - \tau^2) - \tau(\theta_i - \theta_j) + \sqrt{\Delta}\right]^+}{2\tau\theta_i\theta_j(\delta + l + \tau)} \end{cases}, \tag{3.12}$$

and if $\Delta < 0$: $(p_i^*, p_j^*) = (0, 0)$, where $\Delta := \left(\tau(\theta_i + \theta_j) - \lambda\theta_i\theta_j(\delta + l + \tau)^2\right)^2 - 4\tau^2\theta_i\theta_j$. This, along with Proposition 3.1, fully determines the PBE:

Proposition 3.3. *When $\delta > \tau$, the Perfect Bayesian Equilibria (PBE) of the security information sharing game is unique, in which (p_i^*, p_j^*) are provided in* (3.12)*, and $(s_i^*(p_j, n_i), s_j^*(p_i, n_j)) = (0, 0)$ for all $n_i, n_j \in \mathbb{N}^+$ and all $p_i, p_j \in [0, 1)$. That is, both of the firms may invest – to achieve discovery probabilities as given in* (3.12) *– and none of the consequent findings are shared.*

Discussion. When $\delta > \tau$, the competitive gain outweighs the market shrinkage of not sharing the found bugs. Knowing that the found bugs will not be shared, both players, notably even the less efficient player, invest in discovery of the bugs on their own. This is because of two facts: 1- Since the findings are not shared, the firm would be exposed in its bugs if it does not discover and rectify them if it does not invest. 2- Since the other firm invests and expectedly discovers some bugs, the firm will further suffer through the competitive effect of being the sole victim of such bugs if it does not invest.

Comparison to socially optimal outcome:

The social optimal outcome certainly shares the found bugs. Compared to the case of $\delta < \tau$, both players invest strictly more in discovery of the bugs. The social inefficiency of the outcome for the case of $\delta < \tau$ was due to underinvestment. Here, it is primarily due to lack of sharing of the found bugs: if a player would receive information of a

bug that has not discovered itself, the social utility would have improved by preventing the potential direct losses in that player as well as the market shrinkage losses in both players. Another source of social inefficiency is the fact that "both" players make discovery investment: there is a positive probability that the same bug can be discovered independently by both firms. The investment could have been more efficient by preventing such cases of "duplicate effort", if directed to only one player and the subsequent findings are shared. Anther source of social inefficiency, which is again rooted in lack of information sharing of the players, is the possibility of "over-investment" in bug discovery. The optimal expected social utility is the same as was computed in (3.8). Note in particular that it does not depend on the value of δ. Sharing the information in the social optimal removes the competitive effect of δ. However, in the case of $\delta > \tau$, the investment value of both players increases with δ. This means that the threat of competitive losses due to being the sole victim of a security attack can drive both firms to invest inefficiently large values in bug discovery, when they know the discoveries, as competitive advantages, will not be shared. A combination of all of these three effects is responsible for a high social inefficiency in this case.

Comparative Statics

Given $\delta > \tau$ and our example cost functions, we note that players' best response functions as in (3.11) are increasing and concave. Investigating the best-response expressions in (3.11) further reveals:

$$\frac{\partial p_i^{\mathsf{BR}}}{\partial \tau}, \frac{\partial p_i^{\mathsf{BR}}}{\partial l}, \frac{\partial p_i^{\mathsf{BR}}}{\partial \lambda}, \frac{\partial p_i^{\mathsf{BR}}}{\partial \theta_i}, \frac{\partial p_i^{\mathsf{BR}}}{\partial \delta} > 0, \qquad \frac{\partial p_j^{\mathsf{BR}}}{\partial \tau}, \frac{\partial p_j^{\mathsf{BR}}}{\partial l}, \frac{\partial p_j^{\mathsf{BR}}}{\partial \lambda}, \frac{\partial p_j^{\mathsf{BR}}}{\partial \theta_j}, \frac{\partial p_j^{\mathsf{BR}}}{\partial \delta} > 0.$$

This means that player i is willing to invest more as any of the following parameters increases: τ, l, λ, θ_i, and similarly for player j (with θ_i replaced by θ_j). Investigating the effect on the equilibrium point is a bit trickier. For simplicity of exposition, we illustrate the "shift" in the equilibrium pair pictorially. In Figure 3.4, the effect of increasing δ is depicted. Note that, on the "p_i–p_j" plane, $p_i^{\mathsf{BR}}(p_j)$ shifts "up" and $p_j^{\mathsf{BR}}(p_i)$ shifts "right" as the value of δ increases. Hence, the intersection, which indicates the equilibrium, moves towards up and right. The algebraic details of the analysis is removed for brevity. Analysing the effect of each parameter in turn reveals:

$$\frac{\partial p_i^*}{\partial \tau}, \frac{\partial p_i^*}{\partial l}, \frac{\partial p_i^*}{\partial \lambda}, \frac{\partial p_i^*}{\partial \delta}, \frac{\partial p_i^*}{\partial \theta_i}, \frac{\partial p_i^*}{\partial \theta_j} \geq 0, \qquad \frac{\partial p_j^*}{\partial \tau}, \frac{\partial p_j^*}{\partial l}, \frac{\partial p_j^*}{\partial \lambda}, \frac{\partial p_j^*}{\partial \delta}, \frac{\partial p_j^*}{\partial \theta_j}, \frac{\partial p_j^*}{\partial \theta_i} \geq 0.$$

In words, the above inequalities indicate that if any of the following parameters increases, then firms would invest more: τ, l, λ, and δ. Indeed, the higher these parame-

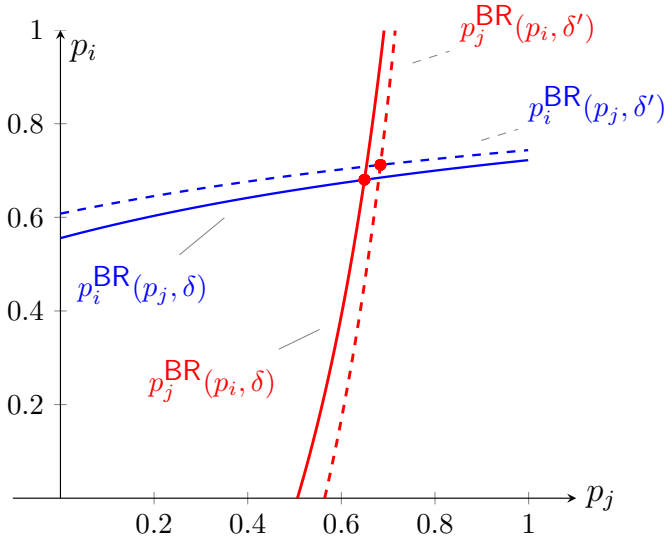


Figure 3.4: Example illustration of the comparative statics for the case of $\delta > \tau$. The parameters used are $\lambda = 1.5$, $l = 0.5$, $\theta_i = 1$, $\theta_j = 0.9, \tau = 0.9$, and the value of δ is increased from $\delta = 1$ to $\delta' = 1.2$. Notice the shift in the equilibrium value towards "up" and "right" as a result.

ters, the more severe impacts of security incidents would be, and thus both firms have to secure themselves, especially when they receive no aid from the other. An interesting result is the effect of improvement in the investment efficiency of the competitor: If θ_j is improved, then firm i invests more in vulnerability research. Intuitively, this is due to the fact that an improvement in the discovery probability of the competitor firm j means more competitive pressure on firm i. This is because each bug that is discovered exclusively by firm j brings it a net advantage of $\delta - \tau$ at the cost of firm i. Thus the increase in efficiency of firm j forces firm i to also improve its probability of discovery, which happens by increasing its investment. This means that the utility of player i decreases as the result of an improvement in player j's efficiency. Specifically, $\partial u_i(p_i^*, p_j^*)/\partial\theta_j < 0$. This is while, $\partial u_j(p_i^*, p_j^*)/\partial\theta_j > 0$. Due to these opposing effects of efficiencies on individual utilities, in general, the equilibrium social welfare, $W(p_i^*, p_j^*)$, which is the sum of the two utilities at the equilibrium, may increase or decrease as θ_i or θ_j is improved. Note, however, that the equilibrium security welfare, $S(p_i^*, p_j^*)$, always improves when θ_i or θ_j increases.

3.5 Mediation: Encouraging Information Sharing

Our analysis in the previous section characterised the players' behaviour in equilibria. For the case of $\delta < \tau$, which pertain to a the case where security acts effectively as a

"common good", sharing of security findings becomes inevitable, and exactly because of that, free-riding emerges, which in turn leads to underinvestment. In contrast, when $\delta > \tau$, which represents cases where security effectively becomes a "competitive advantage", firms would individually strive for their security and refrain from sharing their findings. We observed that none of these outcomes are in line with desirable social planning.

In this section, we make a preliminary attempt to remedy one of the sources of social inefficiency, specifically, failure in information sharing in the "competitive advantage" case. We develop a mediation mechanism that partially removes the negative incentives of sharing the information while allowing the players to gain from its positive effects. Informally put, our mediation plan states that if a firm wants to be informed about n bugs that it failed but the other firm succeeded to discover, it must reveal in exchange n bugs that the other firm is not aware of. Note that this was not possible in the previous sections, as there was no means of making the sharing actions of a firm "contingent" on the action of the other. The mediator effectively ensures that no net "competitive advantage" is lost by sharing the vulnerability findings, as any leakage of an "exclusive" discovery is *matched* by an "exclusive" discovery of the competitor. We will hence refer to our mediation plan as "matched sharing".

Matched sharing operates in two steps: (i) each player/firm submit its set of found bugs to the mediator, along with a specification of a "threshold" as the maximum number of bugs it is willing to exchange with the other firm. (ii) Subsequently, based on the reported sets and the players' thresholds, the mediator moderates the exchange of as many bugs as possible in the following manner: the mediator marks the bugs that are exclusive to each player, i.e., that the other player has not discovered them. Then the information of a bug is transferred from player i to player j iff a) there is an exclusive bug to *match*, i.e., to transfer from player j to i, and b) if the total number of bugs transferred so far does not exceed either one of the players' requested maximum threshold. Note that the mediator is not a strategic player, and its behaviour is known to and trusted by both players.

From the above description, a sharing action of a player entails the selection of the threshold on exchange number. Note specifically, that we can without loss of generality assume that both players submit all of their findings to the mediator.[16] This is because the players can restrict the sharing of their findings by specifying the

[16] Assuming that both parties have established trust with the mediator. Although out of the scope of our work, it is worth mentioning that trust establishment is non-trivial, and may itself be a research problem, e.g., incentive analysis. A potential solution for this might come from cryptography, namely *multi-party computation*. In this case, the two firms can themselves securely simulate the mediator (if the mediator's algorithm is known), so that no trust is needed. This can itself be a topic for future research.

threshold. For instance, no sharing corresponds to requesting a threshold of "zero". Note that due to the nature of the Bayesian game, each player must pick this bound for every realisation of bugs it discovers (given the investment decisions). Formally, we can reuse the notations $s_i(p_j, n_i)$ and $s_j(p_i, n_j)$ to represent the sharing strategies, with the different interpretation that s_i and s_j denote the threshold, i.e., the maximum number of their bugs to be shared by the mediator to the other player. Hence, the expressions in (3.2) in the presence of the mediator and the new interpretation of the strategies become:

$$\mathbb{E}[B_{i,\neg j}|\omega, s_i, s_j] = n_i - n_{ij} - \min\{s_i(p_j, n_i), s_j(p_i, n_j), n_i - n_{ij}, n_j - n_{ij}\}$$
$$\mathbb{E}[B_{\neg i,j}|\omega, s_i, s_j] = n_j - n_{ij} - \min\{s_i(p_j, n_i), s_j(p_i, n_j), n_i - n_{ij}, n_j - n_{ij}\}$$

and, as before, $\mathbb{E}[B_{\neg i,\neg j}|\omega, s_i, s_j] = b - n_i - n_j + n_{ij}$. In words, the term represented by the min function determines the number of bugs that are exchanged between the players, which should be no more than the bounds set by both firms, as well as what each firm individually has to offer. This in turn gives:

$$\begin{aligned} u_i(\omega, \sigma_i, \sigma_j) = &- c_i(p_i) + \delta(n_i - n_j) - \tau(b - n_{ij}) - l(b - n_i) \\ &+ (2\tau + l)\min\{s_i(p_j, n_i), s_j(p_i, n_j), n_i - n_{ij}, n_j - n_{ij}\} \end{aligned} \tag{3.13}$$

As we can see, the only term that involves $s_i(p_j, n_i)$ is the last term. Maximisation of the expected utility of player i given the strategy of player j therefore translates to maximising $\min\{s_i(p_j, n_i), s_j(p_i, n_j), n_i - n_{ij}, n_j - n_{ij}\}$. Hence, we have the following result:

Proposition 3.4. *Suppose $p_i, p_j < 1$. The weakly dominant pure Bayesian Nash Equilibrium of the second stage of the game is $(s_i^*(p_j, n_i), s_j^*(p_i, n_j)) = (n_i, n_j)$ for all $n_i, n_j \in \mathbb{N}^+$ and $p_i, p_j \in [0, 1)$, i.e., asking the mediator to share the maximum number of exclusive bugs. This proposition holds irrespective of the distribution of the total number of bugs, or correlation in the discovery of bugs.*

Proof. First, note that irrespective of the choice of s_j, $s_i(p_i, n_i) = n_i$ maximises the expression $\min\{s_i(p_j, n_i), s_j(p_i, n_i), n_i - n_{ij}, n_j - n_{ij}\}$, and likewise for $s_j(p_i, n_j) = n_j$. Hence $(s_i(p_j, n_i), s_j(p_i, n_j)) = (n_i, n_j)$ for all $n_i, n_j \in \mathbb{N}^+$ and $p_i, p_j \in [0, 1)$ belongs to the set of pure Bayesian Nash equilibria of the second stage of the game. To see the weak dominance, consider the cases where $n_j > n_i > 0$ and $n_{ij} = 0$. Note that $\Pr[N_j > n_i \wedge N_{ij} = 0 \mid N_i = n_i] > 0$. Consider the strategy of player j as $s_j(p_i, n_j) = n_j$ for all $n_j \in \mathbb{N}^+$. Then $u_i(\omega, (p_i, n_i), (p_j, s_j)) > u_i(\omega, (p_i, s_i'), (p_j, s_j))$ for any $s_i'(p_j, n_i) < n_i$, because: $\min\{n_i, s_j(p_i, n_j), n_i - n_{ij}, n_j - n_{ij}\} > \min\{s_i'(p_j, n_i), s_j(p_i, n_j), n_i - n_{ij}, n_j - n_{ij}\}$ for any $s_i'(p_j, n_i) < n_i$ when $n_j > n_i$, $n_{ij} = 0$ and $s_j(p_i, n_j) = n_j$. □

3.5.1 Game's First Stage: Investment in the Presence of the Mediator

Given the weakly dominant equilibrium in Proposition 3.4, $\min\{s_i^*(p_j, N_i), s_j^*(p_i, N_j), N_i - N_{ij}, N_j - N_{ij}\} = \min\{N_i, N_j\} - N_{ij}$. Hence, the utility of player i in (3.13) becomes:

$$\begin{aligned}\mathbb{E}\left[u_i(\omega, p_i, p_j, s_i^*, s_j^*)\right] &= -c_i(p_i) + \delta\mathbb{E}[N_i - N_j] - \tau\mathbb{E}[B - N_{ij}] - ql\mathbb{E}[B - N_i] \\ &\quad + (2\tau + l)(\mathbb{E}[\min\{N_i, N_j\}] - \mathbb{E}[N_{ij}]) \\ &= -c_i(p_i) + \lambda\delta(p_i - p_j) - \lambda\tau(1 - p_i p_j) - \lambda l(1 - p_i) + (2\tau + l)(\mathbb{E}[\min\{N_i, N_j\}] - \lambda p_i p_j)\end{aligned}$$

The term $\mathbb{E}[\min\{N_i, N_j\}]$ depends on the specific distribution of the total number of bugs. A good candidate is the Poisson distribution. The presence of this term in the utility function prevents a closed-form solutions for the best responses and the equilibrium points. Instead, we pictorially illustrate in Figure 3.5 the potential usefulness of the mediator when $\delta > \tau$, i.e., when players are motivated more by competition than aggregate security. Figure 3.5a depicts the equilibrium points of players' investments in two cases: sharing in the absence of the mediator (which leads to no sharing) and our "matched sharing". These are set in the context of low security damage (l) compared to competitive advantage (δ) and inefficient investment ($\theta_i = \theta_j = 0.1$). The end result is that with matched sharing, both players invest more in finding vulnerabilities, which guarantees a better security for both. However, the social welfare, as well as the individual utilities of both players, worsens with the introduction of the matched sharing, as it exacerbates the already inefficiently high investments of the players in this example.

In contrast, Figure 3.5b shows the effect of our mediator plan in situations with either a significant security damage value (large l) or efficient investments (high θ_i, θ_j), or both. In such scenarios, equilibrium points of the two cases are relatively close to each other, i.e., they make similar levels of investments. With the help of the mediator, players would share their intelligences and thus gain extra value in security, making mediation a superior solution to opportunistic sharing. This suggests the potential of our matched sharing mediation scheme, and that it should be in the interest of the social planner to monitor environment parameters and establish trusted mediation among firms whenever appropriate for players/societal benefits.

Remark. In explaining the adverse effect of the mediator to social welfare and individual utilities in Figure 3.5a, we notice that following our model, the expected number bugs commonly discovered by both firms are relatively high (taken straightforwardly from the probabilistic model). As a result, this indicates that matched sharing would not be very useful, since both firms would actually not exchange much information, al-

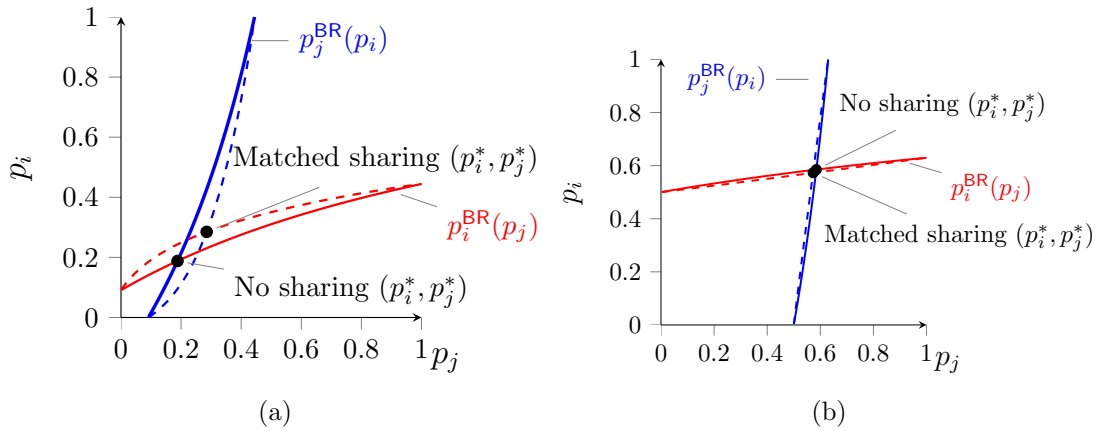


Figure 3.5: Illustration of opportunistic sharing vs. matched sharing when $\delta > \tau$, with $\delta = 10$, $\tau = 1$, $\theta_i = \theta_j = 0.1$, with (a) $l = 1$ and (b) $l = 10$.

though they would be willing to share everything. In reality however, numerous factors might be introduced that affect this figure. For example, large and complex software systems are likely to possess a large number of unknown vulnerabilities, thus reducing the chance for common discoveries. Further, different research teams have different methodologies and procedures, making their chances of uncovering each bug unidentical. All these together would hopefully lessen the expected number of vulnerabilities found by both firms, and thus encouraging the presence of the mediator. Also, efforts in maintaining tight cooperation in the security community (conferences, seminars, discussion groups) might prove useful to eliminating common discoveries of bugs at an early phase of the research procedure.

3.6 Conclusion

In this work, we focused on the problem of sharing cybersecurity information, as an envisioned pillar of cybersecurity planning for a more secure infrastructure. We analysed the strategic decisions of two competing firms with regards to investment for discovery of security vulnerabilities (generating valuable cyber-intelligence) and subsequently, to share their findings. We showed that sharing becomes a dominant strategy when security tends to behaves as a common good, i.e., when the common losses as a result of security attacks outweigh the competitive gains of being protected. We analysed how in turn this leads to free-riding of less efficient firm and the under-investment of the more efficient firm. We also established that when security effectively becomes a competitive advantage, i.e., when there is a net positive gain when a competitor is a sole victim of an attack, then sharing no information becomes the dominant strategy, with negative

implication on the social efficiency. Finally, we provided a monetary-free light-weight mediation mechanism that (partially) enables sharing of the found vulnerabilities in cases where they fail to achieve any sharing on their own.

Future Research This work has the potential to be extended in many directions. We have already made some grounds in extending our results to the multi-player situation. An interesting addition is considering "features" for the found bugs, such as severity (seriousness of the potential damage), sophistication (exploitability), etc., and hence letting the sharing strategies depend on the type of the found bug as well. Investigating the behaviour of risk-averse players – as opposed to risk-neutral in this work – is another problem. Identifying other types of "security information" to share is another interesting direction, for instance, revealing past incidents of successful attacks and resultant losses carries some market implications that sharing merely discovered security vulnerabilities does not. Also, we assumed that both firms use a common implementation (the "platform"). If instead, for instance, the firms are using a common protocol but with their private implementations of it, then "some" of the discovered bugs may be just exclusive to that party's implementation. Sharing found bugs now requires a modified analysis. Investigating other means of encouraging sharing is another important direction. An example is "bargaining": A player starts by sharing one bug, then the other player matches with a bug of its own findings, and so on, until one stops. Another example is a generalisation of the "matched sharing" mechanism in this work by allowing unequal number of matching that may involve some randomisation as well. An exchange market of vulnerabilities is another idea, although it may suffer from adverse selection and moral hazard.

Chapter 4

Optimal Contracts for Outsourced Computations

While expensive cryptographically verifiable computation aims at defeating malicious agents, many civil purposes of outsourced computation tolerate a weaker notion of security, i.e., "lazy-but-honest" contractors. Targeting this type of agents, we develop optimal contracts for outsourcing of computational tasks via appropriate use of rewards, punishments, auditing rate, and "redundancy". Our contracts provably minimise the expense of the outsourcer (principal) while guaranteeing correct computation. Furthermore, we incorporate practical restrictions of the maximum enforceable fine, limited and/or costly auditing, and bounded budget of the outsourcer. By examining the optimal contracts, we provide insights on how resources should be utilised when auditing capacity and enforceability are limited. Additionally, we consider the effect of side-channel information and collusion among contractors. Through careful design of incentives, we demonstrate that it can still be optimal to use multiple contractors even if they are to collude. Finally, we present a light-weight cryptographic implementation of the contracts to mitigate the double moral hazard problem between the principal and the agents.

4.1 Introduction

The idea of outsourcing complex computation tasks has been proposed and implemented in a variety of applications. Research projects involving complex analysis on a huge multitude of data have utilised parallel processing of their computations on the processors of millions of volunteering Internet users. These include search for extra-terrestrial life (*SETI@Home*), investigation of protein folding and computational drug

design (*Folding@Home* and *Rosetta@home*). Businesses from different sections including finance, energy infrastructure, mining and commodities transport, technology and innovation [107] have also realised the benefits of outsourcing (data, computation, etc.) and "moving to the cloud". The cloud, as a dedicated infrastructure with specialised man-force and powerful computing capabilities, along with the ability to pool demands from different clients and dynamic assignment of the resources can reduce the cost of computation. On the other hand, the outsourcer is also relieved from the dedicated investment in its computing infrastructure and in addition, has the total flexibility of pay-per-use paradigm, to flex-on or to flex-off services effortlessly [107]. The growing trend of outsourced computing have made possible small virtualised computers and smart devices with powerful computational power, with applications in critical mission scenarios as well as everyday use.

In all of these scenarios, there is a concern for the outsourcer (client) about the correctness of the returned results. The provider of computation services (the servers) have an economic incentive to return guessed results as opposed to performing the computation completely and honestly, and thereby save on the computation work. Hence, to make this paradigm viable and guarantee soundness of the results, there must be an auditing mechanism in place. The auditing however is not free: it either creates computational overhead for the client, the server, or both. For example, cryptographic verification methods, such as homomorphic encryption [56] and Probabilistically Checkable Proofs (PCPs) [132,134], have been developed (and are being improved upon) that provide a proof of correctness for each and every outsourced computation task. Auditing can also be done through a trusted third party for a fee, say, via *re-computation*. Alternatively, a *redundancy scheme* can be employed in which the same job is outsourced to multiple servers and the results are checked against each other. Here it is important to assume that these servers are totally uncorrelated.

Irrespective of the auditing mechanism, the outsourcer can set an extremely large fine for detected wrong results, and make cheating theoretically impossible. However, in practice, an extremely large fine is a non-credible threat. A more reasonable assumption is a cap on the maximum enforceable fine, with the special interesting case where the cap is zero. In this work we provide a concrete and general approach based on Principal-Agent modelling from game theory to optimal contract designs for outsourcing from the client (principal) to the servers (agents). Specifically, we assume a general maximum enforceable fine, maximum budget, and costly and/or limited auditing rate. We formulate the utilities of both the principal and the agents, as well as essential constraints that guarantee honest computation (incentive compatibility) along with their acceptance of the offer (participation). This allows us to effectively

and systematically compute the optimal contract such that the principal's expense is minimised. Our work hence provides a benchmark enabling meaningful comparison among different deployments of computation outsourcing.

In a further step, we relax the assumption that multiple servers in the redundancy scheme are uncorrelated, and instead address two main threats that previous research neglect to point out: *side-channel* information leakage and *collusion*. In particular, we firstly notice that irrespective of the method of auditing (directly or through redundancy), it is critical that the tasks for which the auditing occurs are not earmarked, or else the contractor would know when it could get away with cheating. In cases where the contractors cannot communicate, this is a good assumption. However, the agents may be able to find a side channel that enables them to find out whether the same task is outsourced to multiple agents for redundancy check or not. Second, the contractors may further collude to report the same guessed result and hence undermine the whole scheme. It is thus questionable whether the outsourcing scheme can still provide any benefit in the face of these two challenges, and if so, how exactly. Specifically, we consider an outsourcer that can use a hybrid of direct auditing and auditing through redundancy. We develop the optimal contracts in closed-form in the presence of a side channel and compare its characteristics with optimal contracts in the absence of such side information. Moreover, we develop two "bounty" schemes and provide sufficient conditions to make redundancy scheme a preferred method over direct auditing even in the presence of collusion.

The chapter is structured as follows: In Section 4.2, we briefly overview previous results in relation to our approach and describe our contributions. This is followed by a detailed motivation of our contract model in Section 4.3, along with descriptions of important constraints that make the problem non-trivial. In Section 4.4, we compute optimal contracts involving only one agent, and explore related improvements. In Section 4.5, we allow the principal to also potentially outsource the same task to multiple non-colluding agents as an alternative means of auditing and develop optimal hybrid contracts. We further establish the global optimality of our hybrid two-agent contracts among all possible contracts involving any number of non-colluding agents with respect to the notion of Nash Equilibria. In Section 4.6, we develop optimal contracts when the outsourcer suspects that the agents may find out (through leakage of information) about whether the same task is sent to another agent. Subsequently in Section 4.7, we focus on the case where the agents potentially collude with each other. We then comment in Section 4.8 the cryptographic implementation of our contracts, i.e., how to enforce the terms and policies in an automated way. This section addresses the problem of *double moral hazard*, in which not only the agents/contractors can cheat,

but also the principal/outsourcer. Finally, in Section 4.9, we conclude the chapter with a summary of the results and remark on some potential future directions.

4.2 Related Work

A line of research is focused on designing reliable verification techniques for outsourcing of special-purpose computations. For instance, [144] investigates outsourcing of linear optimisations. Another notable examples are queries on outsourced databases, including typical queries [7,37] and aggregation [149]. Their main paradigm is for the querier to rely on trusted information directly given by the data owner (outsourcer) to verify the results returned by the servers.

Verification methods for general-purpose computing also appear in several remarkable works. In [99] verification is performed by re-executing parts of the computation. A variation is presented in [31] in which the authors utilise redundancy over multiple agents, assuming that at least one of them is honest. Outsourced computation has also caught attraction in cryptographic research: in a seminal work, the authors of [55] formally define verifiable computation and give a non-interactive solution. Their solution uses Yao's garbled circuits to represent the computation and homomorphic encryption to hide such circuits from the agents. More efficient but interactive solutions that use *probabilistically-checkable proofs* (PCPs) have since been developed such as Pepper [131] and Ginger [133]. All of these verification techniques are, however, costly in terms of computation, memory, incentive rewards, etc., either to the prover or the verifier, or both. For example, the scheme in [99] requires partial re-execution of the tasks, and the verification in [31] incurs cost in the redundancy of the number of computing agents. Also, efficient protocols like Pepper still incurs a cost in the order of m^3 [131] on the principal, where m is the size of the problem. The cost of employing verifiable computing across these different schemes hence raises the important question of how to use them economically, especially when there is a flexibility in parameters that govern the overall cost to the outsourcer.

Incentive-based solutions such as [14,104] have studied contracts that the outsourcer may offer to the agents and through a combination of auditing, fines and rewards, honest computation is enforced. They make use of principal-agent model to design contracts that the agents would accept and become honest. In [14], Belenkiy et al. focus on designing contracts for distributed computing projects with a large pool of agents. They focus on minimising the fine-to-reward ratio as a mean to attract agents/workers, as well as consider the case of irrational/malicious agents. However, their work do not emphasise on ensuring/incentivising complete honesty of agents. Nix and Kantarcioglu,

on the other hand, attempt to design contracts that incentivise complete honesty of agents. They employ redundancy as a mean for detecting cheating agents.

On the other hand, the problems of information leakage and collusion among players have also been studied in the game-theoretic community. Many notable works on information leakage focus on the games with espionage [93]. Here espionage means that a player is able to observe (with noise) the strategy of another before deciding on his own. While Matsui [93] considers repeated games of this form, Solan and Yariv [137] study the normal form game in which a player is able to purchase espionage information. The most well-known concept of collusion in non-cooperative games is *cheap talk* [94]. A game with cheap talk allows players to take part in arbitrary communication with each other before they pick their own strategies. This gives an opportunity for them to coordinate their decisions beforehand. One main problem with cheap talk, as explained by Farrell [51], is that players can lie about their supposed strategy. For example, Croson et al. [41] show that lies and non-credible threats together influence bargaining offers and responses. Further, the whole cheap talk process may itself be a bargain. From another perspective, a subfield of game theory is dedicated to collusion, namely *cooperative* game theory [26]. However, the main goal of cooperative game theory is to form stable coalitional structures, and which is more applicable at societal level problems than in strategic environments.

Regarding the problem of double moral hazard in principal-agent model, [42] suggests that it could be resolved if the principal owning an enterprise can force the agent to purchase that enterprise at prenegotiable price. This serves as a threat to both the principal and the agent, given that at the time of making the efforts, the principal has observed the agents' efforts, but not the eventual profit of the enterprise. Other mechanisms have also been proposed to address this problem in different situations, such as incomplete insurance [39] and money back [92].

Our contributions. Motivated by lack of feasibility in current techniques for verifiable computation, we abstract the verification techniques as an auditing tool with a exogenous cost and provide incentive-based contracts that minimise the expected cost of the principal. Our model can be applied to any special-purpose or generic verification scheme. Our contributions generalise the results in [14, 104] by (1) extending the feasibility of honesty enforcing schemes for *any* bound on the enforceable fines and *any* auditing capacity; (2) explicitly accounting for the cost of auditing and treating the auditing rate as one of the choice variables; and (3) providing optimal contract that minimise the aggregate cost of the principal as a combination of incentive payments and auditing costs. In short, our work extends efficiency of incentive-based solutions

by minimising the expense of the outsourcer. It also extends applicability by employing a general abstraction of verification method that can be captured by the notion of cost and auditing capacity. We also address the problem of leakage of information about task allocation during contract implementation, and that agents can collude in cheating the principal through a simple *bounty hunter* scheme. Finally, we also address the problem of double moral hazard commonly seen in principal-agent models. Our situation is slightly harder than others in economic context, in that the principal has complete advantage in cheating, as its utility can be completely realised before spending any efforts. To resolve this, we make use of cryptographic mechanisms to allow agents detect that the principal has cheated and punish it accordingly.

4.3 Problem Definition: General Setup

In this section, we describe the general setting of the problem and basic assumptions behind our model. A list of notations is provided in Table 4.1 for reference.

The outsourcer, which we refer to as the *principal*[1] has a deterministic *computation task* to be executed to obtain the output (result). Instead of executing the task itself, the principal hires a set of *agents*[2] to do this. The principal aims to enforce *fully honest* computation of the task through setting a contract, involving rewards, auditing, and punishments (fines).

The principal and the agents are each selfish non-cooperative expected utility maximisers. Initially, we assume that everybody is risk-neutral, i.e., they have no strict preference between their expected utility and their utility of expected reward, and hence [58, ch.2.4], their utilities are linear function of the costs (with negative sign) and the rewards (with positive sign). Moreover, we assume that agents are "lazy but not malicious", that is, they do not have any interest in potentially reporting dishonest computations other than saving in their computation cost. Suppose the range and the probability distribution of the computation result is known. Generating a guessed output according to this distribution has zero computation cost and accuracy probability of q_0 (which can be negligibly small if the range of the output is large). For the sake of generality, as in [15], suppose each agent also has access to a private and independent *tricky algorithm* Alg that generates the correct output with probability q_1, where $q_0 < q_1 < 1$, at the cost of $c(q_1) \geq c(q_0) = 0$. The cost of honest computation is $c(1)$, which is strictly greater than $c(q_1)$.[3] To enforce honesty of the agents, the principal

[1] Also called the *boss* [15], *master* [38], *outsourcer* [32], *client* [56], *data owner* [105], etc.

[2] Also referred to as the *workers*, *servers*, *clouds*, or *contractors*.

[3] Using the same method in [15], we will explain at the end of this section how to discourage the use of this tricky algorithm Alg, and thus eliminating the need for q_0 and q_1 in our analysis.

audits the returned result with probability λ. We assume that auditing is perfect, i.e., if the output is indeed correct, the audit definitely confirms it (no "false positives"), and if the output is incorrect, the audit surely detects it (no "false negatives"). In the most basic contract, the principal decides on an auditing rate λ, sets a penalty (fine) f for detected erroneous answers and reward r otherwise. What make the problem non-trivial are the following observations:

1. **Costly detectability of cheating**: that auditing *all* of the results is either *infeasible* or *undesirable*. Regarding the infeasibility, suppose that in the long run the principal has a continuous demand (e.g. the Folding@Home project) of tasks awaiting computation, appearing at a rate ρ tasks per unit time. Also, suppose that each audit takes the principal ν machine cycles, and the computation capacity of the principal's machine is κ cycles per unit time. Then the maximum feasible rate of verification is $\frac{\kappa}{\nu\rho}$.[4] Moreover, auditing (e.g. through re-computation) may be costly as it will consume the computation power of the principal's machine and slow it down, or it will require obtaining additional hardware. The principal chooses the probability of auditing of a task $\lambda \in [0, \Lambda]$, where $0 < \Lambda \leq 1$ is associated with the computational capacity of the principal. The principal incurs the cost $\Gamma(\lambda)$ which is non-decreasing in λ. For simplicity of exposition, we assume a linear relation: $\Gamma(\lambda) = \gamma\lambda$ for a given $\gamma \geq 0$. An alternative to the occasional redoing of the whole computation by the principal can be using a third-party cloud that is highly reliable but costly (with per access cost of γ). For this scenario, the maximum auditing rate Λ is one, i.e., all of the tasks could be audited, albeit at an excessive cost.

2. **Limited enforceability of the fines**: The problem of verifiable computing could become trivial if there is no bound on the fine that can be practically levied on a wrongdoer: as long as there is even a tiniest probability of detection, then the principal can make the expected utility of the smallest likelihood of cheating become negative by setting the fine for erroneous results large enough. The issue with this argument is that such a fine may be extremely large and hence, become an *incredible threat*, in that, if the cheating of an agent is indeed caught, the fine is practically or legally non-collectable. Thus, existence (feasibility) results of honesty enforcement that rely on choosing a "large enough"

[4] Note that even when the principal is verifying at full capacity, it should not pick the next immediate task to verify after finishing the previous one, since it may create a "learnable" pattern of audited tasks, which the agent can use to only be honest when computing them. This however can be avoided if the principal picks uniformly randomly tasks at the rate of $\frac{\kappa}{\nu\rho}$ and store them in a queue. However, the practical buffer has a storage limit. Consequently, the maximum feasible auditing rate with no essential pattern is strictly less than the full capacity rate $\frac{\kappa}{\nu\rho}$.

fine are rather straightforward and uninteresting. In particular, such approaches leave unanswered the question of whether honest computation is still attainable for a bounded enforceable fine below their prescriptive threshold. Moreover, such results do not provide a good metric of comparison between alternative incentive schemes, or across different choices of parameters for a particular scheme. We will explicitly introduce $F \geq 0$ in our model to represent the maximum enforceable fine and obtain the optimal contracts subject to $f \leq F$. This can be the "security deposit", prepaid by the agent to the principal, that is collectible upon a provable detection of an erroneous result. A special case of interest is $F = 0$, i.e., when the only means of punishment is refusal to pay the reward.

3. **Limited budget**: As with the maximum enforceable fine to make it a credible threat, the maximum instantaneous "budget" of the principal leads to a bound on the reward to make it a credible promise. Let the maximum instantaneous payable reward by the principal be R. Thus, we require: $r \leq R$.

In the language of game theory, the contract offering and computation process above can be conveniently captured using the notion of *Stackelberg game* [82]. The general notion of game can be expressed as follows:

Definition 4.1. *Let P denote the principal and A denote the set of agents. Let $\mathcal{O}$ be the a set of possible contract offers, and $\{D_i\}_{i\in A}$ be the set of agents' decision spaces, with $\mathscr{H} \in D_i$ denotes the decision to accept the contract and perform honest computation. Let $\{b_i\}_{i\in A}$ with $b_i : \mathcal{O} \times (\times_{i\in A} D_i) \to \mathbb{R}$ be the set of agents' benefit functions, and $\mathcal{C} : \mathcal{O} \times (\times_{i\in A} D_i) \to \mathbb{R}$ be the principal's cost function such that $\mathcal{C}(o, (d_i)_{i\in A}) = \infty$[5] if some $d_i \neq \mathscr{H}$. With respect to the above, an **outsourced computation** game consists of the following stages:*

1. ***Leader**: the principal P picks a contract offer $o \in \mathcal{O}$, and*
2. ***Followers**: the agents A participate in a strategic-form game $\Gamma_o = \langle A, \{D_i\}_{i\in N'}, \{u_i\}_{i\in A}\rangle$ with $u_i((d_i)_{i\in A}) = b_i(o, (d_i)_{i\in A})$. Suppose the agents select a strategy profile $(d_i)_{i\in A}$, the principal P receives utility $u_P(o, (d_i)_{i\in A}) = -\mathcal{C}(o, (d_i)_{i\in A})$.*

The main objective of this work is to find an optimal contract for the principal, which implies its minimum expense whilst guaranteeing agents' participation and fully honest computation. This can be translated to finding an equilibrium point of the above game involving a principal's offer $o \in \mathcal{O}$ and the agents' strategy profile $\boldsymbol{d} \in \times_{i\in A} D_i$ such that $\boldsymbol{d} = \{\mathscr{H}, \ldots, \mathscr{H}\}$. Indeed, from standard techniques for analysing

[5] This emphasises that the principal strictly aims at having its offer accepted and honestly executed.

Stackelberg games, an effective way to find such equilibrium is to look for a contract offer that yields minimum cost to the principal, given that the contract terms are attractive enough to guarantee agents' participation and honesty. In other words, it means to solve the following optimisation problem:

$$\begin{aligned} &\min_{o \in \mathcal{O}} \ \mathcal{C}(o, \mathscr{H}, \ldots, \mathscr{H}) \\ &\text{s.t. } \boldsymbol{d} = (\mathscr{H}, \ldots, \mathscr{H}) \text{ is a Nash equilibrium of } \Gamma_o.^{6} \end{aligned} \tag{4.1}$$

For the rest of the chapter, we strictly rely on the above optimisation problem to compute the desirable contracts for the principal under various settings.

4.3.1 Eliminating Clever Guesses

An inherent problem of outsourced computation is that the agent can be lazy and cleverly guess (using a tricky algorithm Alg) the result, instead of honestly computing for it. Even worse, if the guess is correct, it is unlikely to be distinguishable from an honestly computed one. For instance, consider the question of whether a large natural number is a prime: the deterministic guess of "no" is most likely correct. When the principal receives an answer, it then performs recomputation to check if a number is prime, then compare with the received answer. Thus, a correct guess would get away with a reward. Since we want to ensure honesty, it is desirable to punish the agent even if the returned output is correct. However, this might be infeasible if the principal cannot bind the correctness of the result to the honesty of the agent.

One way to mitigate the possibility of "clever" guesses is to enlarge the output range by requiring the agent to return not just the final computation output, but also snapshots of intermediate steps of the computing process [15]. Consider the example of computation that checks primarity of a number above. A completely random guess would have success chance $q_0 = 0.5$. A tricky algorithm that always answer "no" would succeed with probability q_1 increasing with the value of the input. If we require as part of the output not just the answer, but also temporary data produced during the primarity test, a completely random guess would have success chance $q_0 = 2^{-b}$ where b is the size of the output in bits. On the other hand, a much more tricky and complicated algorithm Alg' is required to guess both the answer and the temporary data, if the agent wants to keep up with the success probability q_1 as previously. The principal, on the other hand, also needs to perform recomputation and collects temporary data as well as the answer to compare with the agent's report.

[6] When there is only one agent, then the requirement effectively becomes $\mathscr{H} \in \arg\max_{d \in D} u(d)$.

Table 4.1: List of main notations

parameter	definition
λ	probability of auditing an outsourced computation by the principal
Λ	the physical upper-bound on λ
γ	cost of auditing (incurred by the principal)
q	probability of a correct computation by the agent
q_0	the correctness probability of a random guess from the output space
q_1	the correctness probability of a guess from the output space by tricky algorithm Alg
$c(q)$	the expected cost of computation to an agent for the correctness level of q
$c(1), c$	cost of an honest computation to an agent
f	fine collected from agent upon detection of an erroneous computation
F	the maximum enforceable fine
r	reward to the agent for an unaudited or audited and correct computation
R	the maximum feasible reward
z	the reserve utility (a.k.a., fallback utility or aspiration) of the agent
H	auxiliary coefficient defined as $c(1) + z$ (§4.4)
K	auxiliary coefficient defined as $(c(1) - c(q_1))/(1 - q_1)$ (§4.4)
$\mathcal{C}$	the expected cost of the contract to the principal
α	probability of using two agents for the same computation (§4.5.1)
F_0	auxiliary coefficient defined as $c/\Lambda - c$ (Proposition 4.5, §4.5.1)
F_1	auxiliary coefficient defined as $c[c - \gamma]^+/[2\gamma - c]^+$ (Proposition 4.5, §4.5.1)
β	probability of auditing by the principal if the task is assigned to two agents and the returned results are different
ν	probability of auditing by the principal if the task is assigned to two agents and the returned results are the same

As borrowed from [15], we make an assumption that it is practically impossible to design an algorithm Alg' that can help the agent effectively and efficiently guess both the temporary data and the answer accurately [7]. In other words, we assume that for all possible attempts to design a PPT algorithm Alg' for any relevant computation in reality, $q_1 = \epsilon(b)$, where ϵ is a negligible function, and b is the size of the output required. Both q_0 and q_1 are thus assumed to be negligible in b, meaning that we can set b arbitrarily large so that we can neglect q_0 and q_1 altogether. This means that for convenience of our analysis, we may reasonably assume further that $q_0 = q_1 = 0$ for the rest of this chapter. As an extra step, in Section 4.8.1 we will discuss the use of hash function to resolve the case when the size b of output is large, which might be prohibitive for transmission of result. Nevertheless, from now on we assume $q_0 = q_1 = 0$, which suggest us to also assume the worst-case scenario that $c(q_0) = c(q_1) = 0$, that is, the cost of making a totally incorrect guess is zero. As a result, we may also eliminate all agents' strategies of employing any tricky algorithm other than a "completely" incorrect guess, as they yield the same effect. We also denote $c = c(1)$, the cost of honest computation.

4.4 Contracts for Single Agent

In this section, we consider the case where the contract is designed for and proposed to only one computing agent. We provide the optimal contract for the basic model in subsection 4.4.1. In subsection 4.4.2, we investigate what happens if the risk-neutrality assumption of the agents is relaxed. We close the case of single-agent in subsection 4.4.3 by generalising our results to contracts in which the principal is allowed to reward unaudited and verified tasks potentially differently. In Section 4.5, we will investigate the multi-agent case.

In this first type of contracts, we consider an outsourced computation game having the following set of contract offers

$$\mathcal{O} = \{(r, f, \lambda) \mid r \in [0, R] \wedge f \in [0, F] \wedge \lambda \in [0, \Lambda]\}$$

Since there is only one agent, the game Γ_o for $o \in \mathcal{O}$ effectively becomes itself an optimisation problem. Denote the agent by A, we now construct its decision space and utility function. In particular, the agent's action given the parameters $o = (r, f, \lambda)$ of the contract set by the principal, is first whether to accept it, and if so, whether to honestly compute it, or make a guess. This means that we can set the agent's decision

[7] Although theoretically there exist computations whose temporary data can be easily guessed, but we believe that real-world computations which require outsourcing are complex enough for guessing of intermediate step data to be infeasible.

to be $D = \{\perp, 0, 1\}$, that is, the agent is free to either reject ($\perp$) the contract or accept it and be correct with chosen probability 0 or 1, respectively.

4.4.1 Optimal Contract for a Single Agent

The principal chooses the contract by setting the rate of auditing and reward and punishment values, in order to maximise its own utility and ensure *fully honest* computation. Following (4.1), the contract parameter $o = (r, f, \lambda)$ should be chosen such that $1 = \arg\max_D u_A(q)$. This can be broken into two parts: $u_A(1) \geq u_A(0)$ and $u_A(1) \geq u_A(\perp)$. The former guarantees that if the agent accepts the contract, it will perform the computation honestly, and the latter ensures that the agent will accept contract. With respect to the Principal-Agent modelling in game theory (e.g. [58, ch.7] or [122, ch.6]), we will refer to the former as the *incentive compatibility* constraint, which appears as follows:

$$u_A(1) = r - c \geq u_A(0) = (1 - \lambda)r - \lambda f \tag{4.2}$$

The agent accepts the contract if its expected utility is larger than its *reserve utility*, $z \geq 0$.[8] Given incentive compatibility, the latter condition, called *participation constraint*, is hence:[9]

$$u_A(1) = r - c \geq u_A(\perp) = z. \tag{4.3}$$

Given the above requirements and based on the general optimisation problem in (4.1), the *optimal* contract for the case of single agent reduces to the solution of the following:

$$\min_{r,f,\lambda} \mathcal{C} := r + \gamma\lambda \tag{4.4a}$$

$$s.t. \quad r \leq R,\ 0 \leq f \leq F,\ 0 \leq \lambda \leq \Lambda, \tag{4.4b}$$

$$r \geq H,\ r\lambda + f\lambda \geq K \tag{4.4c}$$

[8] The reserve utility (also referred to as the *fall-back utility* or *aspiration wage*) is the minimum utility that the agent aspires to attain or can obtain from other offers. Naturally, $z \geq 0$. Note that an implicit assumption here is that the agent is replaceable by any other agent with the same fall-back utility, i.e., there are many agents available with the same reserve utility. Without this assumption, the agent has negotiation power by refusing the contract knowing that it cannot be replaced. Alternatively, z can be thought as to (exogenously) capture the negotiation power of the agents. This is an assumption we make throughout this work.

[9] Participation constraint is sometimes also called Individual Rationality constraint.

where (4.4c) is derived from (4.2) and (4.3) in which we have used the auxiliary coefficients $H := c + z$ and $K := c$ for brevity. Then:

Proposition 4.1. *With the parameters given in Table 4.1, the contract that enforces honest computation and is accepted by the agent, and minimises the cost of the principal is by setting $f^* = F$ and choosing λ^*, r^* as given by the following:*[10]

$$\gamma \leq \frac{K}{\Lambda^2}: \begin{cases} [\frac{K}{\Lambda} - H]^+ \leq F: & \lambda^* = \frac{K}{H+F},\ r^* = H,\ \mathcal{C}^* = H + \frac{\gamma K}{H+F} \\ [\frac{K}{\Lambda} - R]^+ \leq F < [\frac{K}{\Lambda} - H]^+: & \lambda^* = \Lambda,\ r^* = \frac{K}{\Lambda} - F,\ \mathcal{C}^* = \frac{K}{\Lambda} + \gamma\Lambda - F \end{cases} \tag{4.5}$$

$$\gamma > \frac{K}{\Lambda^2}: \begin{cases} [\sqrt{K\gamma} - H]^+ \leq F: & \lambda^* = \frac{K}{H+F},\ r^* = H,\ \mathcal{C}^* = H + \frac{\gamma K}{H+F} \\ [\sqrt{K\gamma} - R]^+ \leq F < [\sqrt{K\gamma} - H]^+: & \lambda^* = \sqrt{\frac{K}{\gamma}}, r^* = \sqrt{K\gamma} - F, \mathcal{C}^* = 2\sqrt{K\gamma} - F \\ [\frac{K}{\Lambda} - R]^+ \leq F < [\sqrt{K\gamma} - R]^+: & \lambda^* = \frac{K}{R+F},\ r^* = R,\ \mathcal{C}^* = R + \frac{\gamma K}{R+F} \end{cases} \tag{4.6}$$

For $F < [\frac{K}{\Lambda} - R]^+$, the optimisation is infeasible, i.e., there is no honesty-enforcing contract that is also accepted by the agent.

Proof. We present the proof for the case of $\gamma > 0$. The case of $\gamma = 0$ follows more simply. For simplicity, let us fix a feasible fine and compute the solution in terms of f. In the end, we will show that $f = F$ is indeed optimal.[11] We will ignore the constraint of $\lambda \geq 0$ since it is strictly implied by the constraints in (4.4c) (the incentive compatibility). Furthermore, $r \geq H$ implies $r > 0$.

We use the Karush-Kuhn-Tucker (KKT) conditions [11] to solve the above nonlinear (non-convex) programming.[12] Note that our cost and constraint functions are all continuously differentiable. We first use the Mangasarian–Fromovitz constraint qualification (MFCQ) to establish that any minimum must satisfy the KKT conditions, i.e., KKT are necessary conditions of optimality. In the absence of equality constraints, the MFCQ condition means that the gradients of the active inequality constraints are positive-linearly independent at optimum points. For reader's convenience, we summarise the notion of KKT and MFCQ in Appendix A.

In the special case of $R = H$, the constraints $r \geq H$ and $r \leq R$ imply $r = H = R$, and hence, the optimisation problem can be rewritten with only λ as a variable, which is simple to analyse. When $R > H$, only one of the constraints $r - R \leq 0$ and $H - r \leq 0$ is ever active. We will investigate them one at a time. The gradients of the other inequality constraints $r - R \leq 0$, $\lambda - \Lambda \leq 0$ and $K - r\lambda - f\lambda \leq 0$ are respectively:

[10] The notation $x^+ := \max\{0, x\}$.

[11] Alternatively, the following simple argument shows from the beginning that f must be at its maximum value F: Note that the principal can increase the auditing rate λ, or reward r or the fine f in order to enforce the incentive compatibility constraint. Of these three variables, only increasing the fine is costless to the principal.

[12] The nonconvexity arises due to the second inequality in (4.4c).

$(1,0)$, $(0,1)$ and $(-\lambda, -r-f)$. Note that only for $f = K/\Lambda - R$, the last three inequalities can be all active. For this case, the domain of feasible solutions reduces to the singleton point of $r = R$, $\lambda = \Lambda$. For $f < K/\Lambda - R$, no feasible solution exists. For all other cases, at most two of the constraints are active at a time, whose gradients can never be linearly dependent: $(1,0)$ and $(0,1)$ are clearly linearly independent, and both elements of $(-\lambda, -r-f)$ are strictly negative, hence it is linearly independent from each of the other two. Now, consider the $H - r \leq 0$ constraint whose gradient is $(-1,0)$. Note that three of the constraints may be simultaneously active, but their gradient will not be positive-linearly dependent, because both elements of $(-\lambda, -r-f)$ are strictly negative. Hence, the MFCQ normality condition holds.

To systematically obtain the KKT conditions, we introduce the dual multipliers μ_1, μ_2, μ_3 and μ_4, and transform the problem in (4.4) as follows:

$$\min_{r,f,\lambda,\mu_i} \bar{\mathcal{C}} = r + \gamma\lambda + \mu_1(r-R) + \mu_2(\lambda - \Lambda) + \mu_3(H-r) + \mu_4(K - f\lambda - r\lambda)$$

$$\begin{aligned}
\text{s.t.:} \quad \text{primary feasibility:} \quad & r \leq R,\ \lambda \leq \Lambda,\ r \geq H,\ r\lambda + f\lambda \geq K && \text{(4.7a)}\\
\text{duality feasibility:} \quad & \mu_1, \mu_2, \mu_3, \mu_4 \geq 0, && \text{(4.7b)}\\
\text{complementary slackness:} \quad & \mu_1(r-R) = 0,\ \mu_2(\lambda - \Lambda) = 0, && \text{(4.7c)}\\
& \mu_3(H-r) = 0,\ \mu_4(K - f\lambda - r\lambda) = 0. && \text{(4.7d)}
\end{aligned}$$

The first order conditions of optimality are:

$$\frac{\partial \bar{\mathcal{C}}}{\partial r} = 0 \Leftrightarrow \mu_4\lambda = 1 + \mu_1 - \mu_3, \qquad \frac{\partial \bar{\mathcal{C}}}{\partial \lambda} = 0 \Leftrightarrow \mu_4 r = \gamma + \mu_2 - f\mu_4. \tag{4.8}$$

The full solution as in the proposition with F replaced by f is now derived by straightforward investigation of the above conditions. This can be done algorithmically by exhausting the possibilities in the complementary slackness. For readers' interest we provide the code to do this in Mathematica, which is given in Appendix B. This same code is used for all optimisation problems in this chapter. The proof then concludes by noting that the cost such found is strictly decreasing in f, and hence $f^* = F$. □

Discussion. The first observation is that the optimal contract should fully utilise the maximum enforceable fine and punish at no less than F. For large values of enforceable fines, we note that r^* is at H, the minimum value to ensure participation, and $\lim_{F\to\infty} \lambda^* = 0$, which yields $\lim_{F\to\infty} \mathcal{C}^* = H$. These are compatible with intuition as a huge fine implies that honesty can be enforced with minimum compensation and minuscule rate of inspection. When auditing is cheap ($\gamma \leq K/\Lambda^2$), increasing the auditing rate is the better option to compensate for lower values of F to maintain incentive com-

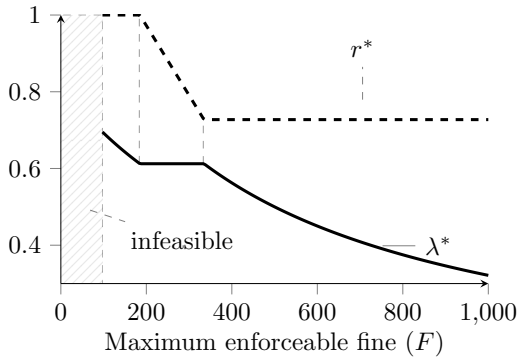


Figure 4.1: Change of contract parameters r^*, λ^* w.r.t. the maximum enforceable fine F (Prop. 4.1, case of $\gamma > \frac{K}{\Lambda^2}$), where $K = 450$, $\gamma = 1200$, $\Lambda = 0.7$, and $c = 400$.

patibility (honest computation). This is unless the auditing rate is at its maximum Λ, in which case, reward must increase above H to maintain incentive compatibility and compensate for the low value of F. Note that in this case, the participation constraint is not active and is satisfied with a slack, while the incentive compatibility constraint is satisfied tightly. For yet lower values of enforceable fine F, even maximum reward $r = R$ and auditing rate $\lambda = \Lambda$ might not impose a strong enough threat against cheating, hence the infeasibility region. When auditing is expensive ($\gamma > K/\Lambda^2$), in order to retain incentive compatibility in the situation of very low fine F, the principal should increase reward, and only consider more frequent auditing if the reward budget R has been reached. Figure 4.1 depicts the optimal parameters of the contract versus the maximum enforceable fine for the latter case ($\gamma > K/\Lambda^2$).

Note that the infeasible region does not necessarily exist. Specifically, when the principal's instantaneous budget R is larger than K/Λ, then there is always a feasible contract. Then even for $F = 0$, i.e., no enforceable fine, a contract that enforces honest computing is feasible, albeit by using high values of reward and/or auditing rate. In such cases, the principal "punishes" audited erroneous computations only through not rewarding the agent. However, it is clear that honesty cannot be enforced with zero auditing rate, and hence the case of $\Lambda = 0$ trivially leads to infeasibility. Moreover, to satisfy the participation constraint at all, R has to be at least as large as H. Hence, for $R < H$, likewise, there exists no feasible contract for any F. We also show that except for the special case of $\gamma = 0$, the optimal contract has the feature that it is *unique*. Figures 4.2a and 4.2b depict the change in the structure of the optimal contract versus varying auditing cost γ and the maximum auditing capacity, respectively. From Figure 4.2a, we can see that for larger values of γ, the optimal contract utilises lower values of inspection rate λ^* while using higher values of reward r

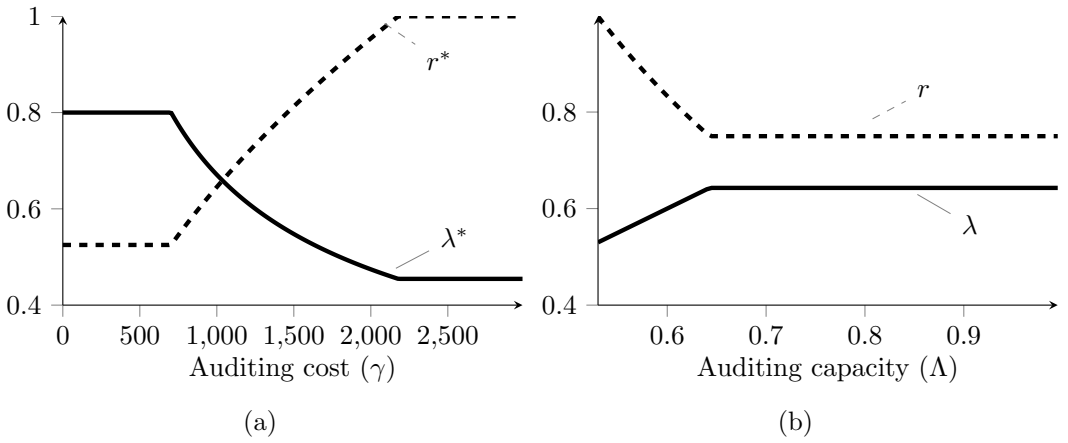


Figure 4.2: Optimal contract parameters w.r.t (a) the auditing cost γ, with $K = 450$, $\Lambda = 0.8$, $c = 400$, and (b) auditing capacity Λ, with $K = 450$, $\gamma = 450$, $c = 450$.

to enforce honest computation. This transition progress culminates when the payment reaches its threshold R, after which the contract remains unchanged. In contrast, Figure 4.2b shows how increasing the maximum auditing capacity affects the optimal contract in the opposite trend: as the principal is more capable of auditing, it should consider more frequent auditing and lessen the reward for honest computation. The payment, however, can never be lowered below H to maintain participation.

4.4.2 A Risk-Averse Agent

So far, we modelled the agent as risk-neutral, i.e., one that is indifferent between its expected utility and utility of expectation, leading to a linear utility function. However, empirically, individuals tend to show risk-aversion regarding decisions that affect their income. By definition, (strict) risk aversion is (strict) preference of expected utility over utility of expectation. Following *Jensen*'s inequality, this is equivalent to assuming a (strictly) concave utility function (ref. e.g. [58, ch.2.4]). We have the following simple but re-assuring result:

Proposition 4.2. *The optimal contract given in Proposition 4.1 developed for a risk-neutral agent stays feasible for any risk-averse agent as well.*

Proof. Assume that the agent values its utility u_A by a (strictly) concave function u. The only two constraints in the optimal contract that may change are the incentive compatibility and participation: (4.2), (4.3). The new participation constraint is: $u(r - c) \geq u(z)$. Due to the increasing property of $u(\cdot)$, this new constraints translates back to $r - c \geq z$, hence no change here.

For analysing the new incentive-compatibility constraint, let us represent the mixed action of the agent by $\mathcal{L}(x, y, 1-x-y)$ which means making a random guess with probability x, using the tricky algorithm with probability y, and doing the honest computation with probability $1-x-y$. With a slight abuse of notation, let $X[\mathcal{L}]$ be the random variable representing the utility of the agent given its mixed action $\mathcal{L}$. Then the risk-neutral incentive compatibility constraint as given in (4.2) is ensuring that $\mathbb{E}[X[\mathcal{L}(0,0,1)]] \geq \mathbb{E}[X[\mathcal{L}(x,y,1-x-y)]]$. Because $u(\cdot)$ is increasing, this inequality implies: $u(\mathbb{E}[X[\mathcal{L}(0,0,1)]]) \geq u(\mathbb{E}[X[\mathcal{L}(x,y,1-x-y)]])$. Further, following Jensen's inequality, since u is concave, $u(\mathbb{E}[X[\mathcal{L}(x,y,1-x-y)]]) \geq \mathbb{E}[u(X[\mathcal{L}(x,y,1-x-y)])]$. Note that $X[\mathcal{L}(0,0,1)]$ is a deterministic random variable (specifically, payoff of $r-c(1)$ w.p. one). Hence: $u(\mathbb{E}[X[\mathcal{L}(x,y,1-x-y)]]) = \mathbb{E}[u(X[\mathcal{L}(x,y,1-x-y)])]$. Therefore, we have shown that (4.2) implies: $\mathbb{E}[u(X[\mathcal{L}(x,y,1-x-y)])] \geq \mathbb{E}[u(X[\mathcal{L}(x,y,1-x-y)])]$, which is the incentive compatibility constraint for a risk-averse agent. □

Note that even though the feasibility of our contract is guaranteed, its optimality might no longer hold. This is because a lower value of fine and/or rewards could potentially maintain incentive compatibility, as intuitively, cheating with a chance of getting caught can be seen as a lottery. However, because the level of risk-averseness of an agent is unknown, we argue that it is best practice to design the optimal contract for the worst case with respect to risk, i.e., risk neutrality. Specially, if a contract is designed assuming a particular degree of risk-aversion of the agent but the agent turns out to be less risk-averse than assumed, then the incentive-compatibility for honest computation may be violated, failing the principal's intolerance of erroneous computations. Accordingly, for the rest of this chapter, we will retain risk-neutrality for agents.

4.4.3 Optimal Contract for a Single Agent: Two-Level Reward

In our contracts so far, verified correct results and unaudited results are rewarded identically at r. Suppose, alternatively, that the principal rewards r_0 for accepted but not audited results and r_1 for corroborated correct answers, and as before, penalises f for detected wrong computations. This way, the principal may hope to save significantly by, for example, not paying for unaudited computations. The new incentive compatibility and participation constraints are: $(1-\lambda)r_0 + \lambda r_1 - c \geq (1-\lambda)r_0 - \lambda f$ and $(1-\lambda)r_0 + \lambda r_1 - c \geq 0$, respectively. The optimisation of (4.4) for a contract with two-level reward changes to:

$$\min_{r_0, r_1, f, \gamma} \mathcal{C} := r_1\lambda + r_0(1-\lambda) + \gamma\lambda$$

$$s.t. \quad r_0, r_1 \leq R,\ f \leq F,\ 0 \leq \lambda \leq \Lambda,\ r_1\lambda + r_0(1-\lambda) \geq c,\ r_1\lambda \geq c - f\lambda.$$

Proposition 4.3. *For $F \geq [c/\Lambda - R]^+$, the optimal single-agent contract for two-level rewarding is given as: $f^* = F$, $\lambda^* = c/(F+R)$, $r_1^* = R$, $r_0^* = Fc/(R-c+F)$, $\mathcal{C}^* = c\,(1 + (\gamma + c - R)/(F+R))$. For $F < [c/\Lambda - R]^+$, the contract is infeasible.*

Proof. The solution can be found by first verifying the MFCQ requirement, then construction of KKT conditions, and finally solving these conditions using exhaustive search to find the optimal regions. While we omit presentation of the last step as it involves a large search, we detail the first two steps, and refer to the appendix for a Mathematica program for searching the KKT conditions. For this we note that it is obviously best to set $f = F$, and assume this from now on. Thus the minimisation problem has only three variables r_0, r_1 and λ.

We verify the MFCQ conditions in the following. When $R = c$, it is obvious that $r_0 = r_1 = c$, and hence there is only one variable left, i.e., λ, therefore the problem becomes trivial. When $R > c$, the gradients of all inequalities on r_0, r_1 and λ are

$$v_1 = \nabla(r_0 - R) = (1,0,0);\ v_2 = \nabla(r_1 - R) = (0,1,0);\ v_3 = \nabla(-\lambda) = (0,0,-1);$$
$$v_4 = \nabla(\lambda - \Lambda) = (0,0,1); v_5 = \nabla(c - r_0(1-\lambda) - r_1\lambda) = (\lambda - 1, -\lambda, r_0 - r_1);$$
$$v_6 = \nabla(c - f\lambda - r_1\lambda) = (0, -\lambda, -f - r_1)$$

We now show the MFCQ requirement: there exist no $a_i \geq 0$ with $i \in \overline{1,6}$, not all zero, such that $\sum_{i=1}^{6} a_i v_i = \mathbf{0}$, in that $a_i > 0$ implies the i-th inequality is active. We proceed by going through each vector, and rule them out of the potential linear dependencies one-by-one, until none is left. Consider $a_1 > 0$ which implies $r_0 = R$, then to form linear-dependency, we need $a_5 > 0$. Consequently, due to the negative value $-\lambda$ of the second component of v_5, we need either $a_2 > 0$ or $a_6 > 0$ to form a dependency. However, the second component of v_6 is also negative, therefore it is required that $a_2 > 0$, which implies $r_1 = R$. This, together with $r_0 = R$, contradict with the fact that $c = r_0(1-\lambda) - r_1\lambda = R$ (due to $a_5 > 0$) because we assume earlier that $R > c$. This concludes that $a_1 = 0$, and we thus ignore v_1.

If $a_2 > 0$, then to form dependencies we need either $a_5 > 0$ or $a_6 > 0$. The former cannot be satisfied, because it would require $a_1 > 0$, thus contradicting with our conclusion above. In the latter $a_6 > 0$, along with others, eventually implies $c = \Lambda(R+F)$. In that situation, we have the incentive-compatibility constraint satisfied only when $r_1 = R$ and $\lambda = \Lambda$, and thus the problem becomes trivial. Assuming $c < \Lambda(R+F)$ nullifies the chance that $a_6 > 0$, which consequently leads to setting $a_2 = 0$, since $a_2 > 0$ leads to no positive-linear dependency.

Next we set $a_3 > 0$, which implies $\lambda = 0$. This is contradict with the incentive-compatibility constraint, thus we must have $a_3 = 0$. For $a_4 > 0$, we need either $a_5 > 0$ or $a_6 > 0$. If we set $a_5 > 0$, either we need both $\lambda = 1$ and $\lambda = 0$ since there is no other available vectors that can form a dependency with v_5, which is also not possible. In case $a_6 > 0$, we then need $\lambda = 0$, which does not satisfy incentive-compatibility. Thus it also holds that $a_4 = 0$. It is then trivial to show that v_5 and v_6 cannot be positive-linearly dependent.

Given that the MFCQ requirement is satisfied, an exhaustive search in solving KKT conditions would yield global minimums for our contract optimisation. The conditions can be expressed as follows:

$$\begin{aligned}
\min_{r,f,\lambda,\mu_i} \bar{\mathcal{C}} &= r_1\lambda + r_0(1-\lambda) + \gamma\lambda + \mu_1(r_0 - R) + \mu_2(r_1 - R) + \mu_3(-\lambda) \\
&\quad + \mu_4(\lambda - \Lambda) + \mu_5(c - r_0(1-\lambda) - r_1\lambda) + \mu_6(c - f\lambda - r_1\lambda) \\
\text{s.t.:} \quad \text{primary feasibility:} \quad & r_0, r_1 \leq R,\ 0 \leq \lambda \leq \Lambda, && \text{(4.10a)} \\
& c - r_0(1-\lambda) - r_1\lambda,\ c - f\lambda - r_1\lambda \leq 0 && \text{(4.10b)} \\
\text{duality feasibility:} \quad & \mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6 \geq 0, && \text{(4.10c)} \\
\text{complementary slackness:} \quad & \mu_1(r_0 - R) = 0,\ \mu_2(r_1 - R) = 0,\ \mu_3(-\lambda) = 0, && \text{(4.10d)} \\
& \mu_4(\lambda - \Lambda) = 0,\ \mu_5(c - r_0(1-\lambda) - r_1\lambda) = 0, && \text{(4.10e)} \\
& \mu_6(c - f\lambda - r_1\lambda) = 0. && \text{(4.10f)}
\end{aligned}$$

The first order conditions of optimality are:

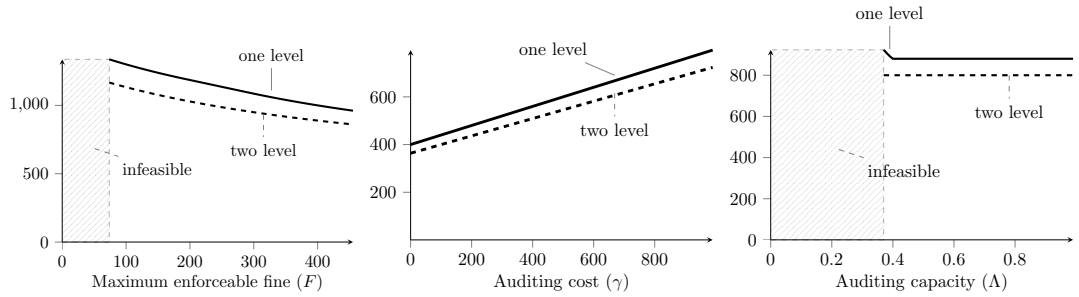
$$\frac{\partial \bar{\mathcal{C}}}{\partial r_0} = 0 \Leftrightarrow 1 - \lambda + \mu_1 - \mu_5(1-\lambda) = 0 \tag{4.11}$$

$$\frac{\partial \bar{\mathcal{C}}}{\partial r_1} = 0 \Leftrightarrow \lambda + \mu_2 - \mu_5\lambda - \mu_6\lambda = 0 \tag{4.12}$$

$$\frac{\partial \bar{\mathcal{C}}}{\partial \lambda} = 0 \Leftrightarrow (r_1 - r_0)(1 - \mu 5) + \gamma - \mu_3 + \mu_4 - F\mu_6 - r_1\mu_6 = 0 \tag{4.13}$$

Solving the above would give the solution as in the proposition statement. □

Discussion of the two level reward contract. First, note that there is no improvement in terms of the infeasibility region compared with the single-level reward contract. However, the achieved cost is always better. This was to be expected as the single-level rewarding can be thought of as a special case of two-level. However, the behaviour of the optimal contract now does not depend on the value of the auditing cost γ. This is where the strength of the two-level rewarding lies: for high values of γ, the two-level contract increasingly outperforms the single reward-level contract.



(a) (b) (c)

Figure 4.3: Optimal contract expense with (a) $c = 400$, $\Lambda = 0.7$, $\gamma = 1200$, $R = 500$, (b) $c = 400$, $\Lambda = 0.7$, $R = 500$, $F = 600$, and (c) $c = 400$, $\gamma = 1200$, $R = 500$, $F = 600$.

Note that the optimal reward for audited and correct results r_1 is at the principal's maximum budget R irrespective of the value of F. The value of reward for unaudited results r_0 is always strictly less than c, i.e., the cost of honest computation (and hence strictly less than r_1 as well). The value of r_0, unlike r_1, depends on F: For higher values of maximum enforceable fine, in fact somewhat unexpectedly, the optimal contract chooses increasing values of reward r_0^*. Still intuitively, a larger threat allows less necessity for auditing, and thus the contract starts to behave as a "lottery", in which the low-chance "winner" receives $r_1^* = R$ and the "loser" $r_0 < c < R$. For completeness, we visualise in Figure 4.3 the comparison between one-level and two-level reward in terms of optimal contract expense of the principal.

4.5 Optimal Contracts for Multiple Agents

When there are more than one agent available, the set of possible contracts gets extended. Specifically, as e.g. [15] and [105] discuss, the principal has the option of submitting the same task to multiple agents and comparing the outcomes. We will refer to this option as the *redundancy* scheme. If the returned results do not match, it is clear that at least one agent is cheating. Furthermore, as [105] assumes, if the agents are non-colluding, and returning the intermediate steps along with the computation result is required, then the probability that the results produced by cheating will be the same will be negligible, which we again assume to be zero (for simplicity). Hence, the returned results are correct *if and only if* they are the same.

In the next subsection, we develop optimal contracts considering two agents. Subsequently, we establish the global optimality of two-agent contracts among any number of agents with respect to the notion of Nash Equilibrium.

4.5.1 Optimal Contracts for Two Agents

Consider the case that there are two agents available $A = \{1, 2\}$. As in the single-agent case, consider a principal that has a computation task and a maximum auditing rate of Λ. Then, in general, a principal can use a hybrid scheme: it may choose to send the same job to both of the agents sometimes, and otherwise to one randomly selected agents. Sending the same task to two agents provides a definite verification, however, at the cost of paying twice the reward, since both agents must be rewarded for honest computation. Hence, an optimal choice of redundancy scheme is not immediately clear, even less so if this schemes is randomised with just choosing one agent and doing independent audits. In this section, we investigate optimal contracts among all hybrid schemes.

As the first step, we construct the set of possible contracts the principal P may offer to the agents. Let $\alpha \in [0, 1]$ be the probability that the principal utilises the redundancy scheme, and hence with probability $1 - \alpha$ it employs only one of the agents (selected equally likely)[13]. If the principal chooses both agents to assign the computation, then it will reward both agents r if the returned results are identical. Otherwise it with punish both agents a fine f. In case only one agent if employed, then P audits with probability ρ. Since auditing only occurs when a single agent receives the task, the likelihood that the task will ever be audited is $\lambda = \rho(1 - \alpha)$. If we consider α and λ as parameters of the contract, then it must be the case that $\alpha + \lambda \leq 1$. Also, because we account for double rewarding, thus the maximum reward r for each agent must be no more than $R/2$. This creates the following set of possible contracts:

$$\mathcal{O} = \{(r, f, \alpha, \lambda) \mid r \in [0, R/2] \wedge f \in [0, F] \wedge \alpha \in [0, 1] \wedge \lambda \in [0, \Lambda] \wedge \alpha + \lambda \leq 1\}$$

To form the agents' behaviours and the game Γ_o, we firstly assume that neither there is any collusion between, nor any communication. Therefore, on the event that any of the agents receives a task, it has no information about the busy/idle state of the other agent. Excluding the option $\perp$ to reject the offer, each agent must choose to be honest with probability $q \in [q_0, 1]$ with $q_0 = 0$ due to our argument in subsection 4.3.1. This makes players' decision spaces $D_1 = D_2 = D = \{\perp\} \cup [0, 1]$, where we denote $0 \in D$ by $\mathscr{C}$ and $1 \in D$ by $\mathscr{H}$. We then assume that if either of the agents rejects the offer, then no contract is signed, and thus their utilities are 0, that is, $u_i(\perp, \cdot) = u_i(\cdot, \perp) = 0$ for $i \in \{1, 2\}$. Otherwise, let $d_1, d_2 \in [0, 1]$ be "non-rejecting" strategies of agent 1 and 2, respectively. The utilities of the agents can be expressed as follows:

[13] We will formally show through the proof of proposition 4.6 that equal randomisation is the best option. Intuitively, this removes any information that the agents may infer upon receiving a task.

Proposition 4.4. *Let $o = (r, f, \alpha, \lambda)$ be a two-agent contract and $d_1, d_2 \in [0, 1]$ be the strategies of the agents, then their utilities are:*

$$u_i(d_i, d_{-i}) = \frac{1}{2}(r + (2(-1 + d_i d_{-i})f + (-1 + 2d_i d_{-i})r)\alpha - cd_i(1 + \alpha) + (-1 + d_i)(f + r)\lambda)$$

Proof. The utility of agent i comprises two parts: when it is the only employed agent, and when both agents are employed. They occur with probabilities $(1 - \alpha)/2$ and α, respectively. Let $\rho = \lambda/(1 - \alpha)$ be the probability that the agent will be audited given that it is the only employed agent, then agent i's expected utility for former case is:

$$\begin{aligned}
&\frac{1-\alpha}{2}\left((1-(1-d_i)\rho)r - (1-d_i)\rho f - cd_i\right) \\
&= \frac{1-\alpha}{2}\left(\left(1-(1-d_i)\frac{\lambda}{1-\alpha}\right)r - (1-d_i)\frac{\lambda}{1-\alpha}f - cd_i\right) \\
&= \frac{1}{2}(cd_i(-1+\alpha) + (-1+d_i)f\lambda - r(-1+\alpha+\lambda-d_i\lambda))
\end{aligned}$$

Similarly, agent i's expected utility for the latter case is

$$\alpha(d_i d_{-i} r - (1 - d_i d_{-i})f - cd_i)$$

Summing the above two terms would give agent i's net utility as in the proposition statement. □

An important point about the expression of $u_i(d_i, d_{-i})$ in the above proposition is that u_i is linearly in d_i. We then recall from our basic optimisation problem (4.1) that we want $(\mathscr{H}, \mathscr{H})$ to be a Nash equilibrium, meaning that $u_i(\mathscr{H}, \mathscr{H}) \geq u_i(\perp, \cdot)$, and that $\mathscr{H} = \arg\max_{d_i \in [0,1]} u_i(d_i, \mathscr{H})$. Note that it is also possible that $(\mathscr{C}, \mathscr{C})$ is another equilibrium, but such is a rather uninteresting because both agents would be punished with higher chance, and thus receive worse utility than with $(\mathscr{H}, \mathscr{H})$, both individually and socially. Due to the linearity of u_i in d_i, the latter is satisfied when $u_i(\mathscr{H}, \mathscr{H}) \geq u_i(\mathscr{C}, \mathscr{H})$. Following the above proposition, the values of these two sides are :

$$u_A(\mathscr{H}, \mathscr{H}) = \frac{1+\alpha}{2}(r - c), \qquad u_A(\mathscr{C}, \mathscr{H}) = (1 - \alpha - \lambda)\frac{r}{2} - (\alpha + \frac{\lambda}{2})f.$$

Assume for simplicity that agents' reserve utility is $u(\perp, \cdot) = z = 0$, we have the participation and incentive compatibility constraints respectively as

$$r - c \geq 0, \qquad r \geq (1 + \alpha)c/(\lambda + 2\alpha) - f.$$

On the other hand, given that agents' strategy profile is $(\mathscr{H}, \mathscr{H})$, the expected cost of the contract to the principal is:

$$\mathcal{C} = 2r\alpha + \gamma\lambda + r(1-\alpha) = (1+\alpha)r + \gamma\lambda.$$

Therefore, the optimal contracts for two agents that make $(\mathscr{H}, \mathscr{H})$ an equilibrium can be found by soling the following optimisation problem, as given in Proposition 4.5:

$$\min_{r,f,\alpha,\lambda} \mathcal{C} := r(1+\alpha) + \gamma\lambda \text{ subject to:}$$
$$r \le R/2,\ f \le F,\ 0 \le \lambda \le \Lambda,\ \lambda \le 1-\alpha,\ \alpha \ge 0,\ r \ge c,\ r \ge \frac{c(1+\alpha)}{\lambda+2\alpha} - f.$$

Proposition 4.5. *Let $F_0 = c/\Lambda - c$ and $F_1 = c[c-\gamma]^+/[2\gamma - c]^+$,[14] the optimal one-level reward two-agent contract that makes $(\mathscr{H}, \mathscr{H})$ a Nash equilibrium is:*

$$\begin{cases} F_1 \le F: & f^* = F,\ \alpha^* = \dfrac{c}{2F+c},\ \lambda^* = 0, r^* = c,\ \mathcal{C}^* = c(1 + \dfrac{c}{2F+c}) \\ F_0 \le F < F_1: & f^* = F,\ \alpha^* = 0,\ \lambda^* = \dfrac{c}{c+F},\ r^* = c,\ \mathcal{C}^* = c(1 + \dfrac{\gamma}{F+c}) \\ F < \min(F_0, F_1): & f^* = F,\ \alpha^* = \dfrac{c - \Lambda(c+F)}{c+2F},\ \lambda^* = \Lambda,\ r^* = c,\ \mathcal{C}^* = \dfrac{c(c+F)(2-\Lambda)}{c+2F} + \gamma\Lambda \end{cases}$$

Proof. Similar to the optimisation of previous contracts, we use the KKT conditions to find the optimal contract for two agents. For simplicity we assume that $f = F$, as this does not affect the optimal solution. Then, we employ dual multipliers μ_i, for $i = \overline{1,7}$, such that

$$\begin{aligned} \min_{r,\lambda,\alpha,\mu_i} \bar{\mathcal{C}} =& r(1+\alpha) + \gamma\lambda + \mu_1(r-R) + \mu_2(c-r) \\ &+ \mu_3(c(1+\alpha) - (f+r)(\lambda+2\alpha)) + \mu_4(-\alpha) \\ &+ \mu_5(-\lambda) + \mu_6(\lambda-\Lambda) + \mu_7(\lambda-(1-\alpha)) \end{aligned}$$

$$\begin{aligned} \text{s.t.:} \quad \text{primary feasibility:} \quad & c \le r \le R,\ 0 \le \lambda \le \Lambda,\ \alpha \ge 0,\ \lambda \le 1-\alpha, \\ & c(1+\alpha) \le (f+r)(\lambda+2\alpha) \\ \text{duality feasibility:} \quad & \mu_1, \mu_2, \mu_3, \mu_4 \ge 0, \\ \text{complementary slackness:} \quad & \mu_1(r-R) = 0,\ \mu_2(c-r) = 0, \\ & \mu_3(c(1+\alpha) - (f+r)(\lambda+2\alpha)) = 0,\ \mu_4(-\alpha) = 0, \\ & \mu_5(-\lambda) = 0,\ \mu_6(\lambda-\Lambda) = 0,\ \mu_7(\lambda-(1-\alpha)) = 0. \end{aligned}$$

[14] We adopt the convention that $x/0 = +\infty$ for $x > 0$.

Along with these are the stationarity conditions:

$$\frac{\partial \bar{\mathcal{C}}}{\partial r} = 0 \Leftrightarrow 1 + \alpha + \mu_1 - \mu_2 - \mu_3(\lambda + 2\alpha) = 0$$
$$\frac{\partial \bar{\mathcal{C}}}{\partial \lambda} = 0 \Leftrightarrow \gamma - \mu_3(f + r) - \mu_5 + \mu_6 + \mu_7 = 0$$
$$\frac{\partial \bar{\mathcal{C}}}{\partial \alpha} = 0 \Leftrightarrow r - \mu_3(2f + 2r - c) - \mu_4 + \mu_7 = 0$$

Solving the system of these equalities and inequalities give us the solution as expressed in the proposition. The last step is to prove that this solution is the global optimal. Like in the previous contracts, our optimisation problem is not convex, therefore, we rely on the MFCQ conditions to satisfy for all local minimisers. Recall that $R \geq c > 0$, $\gamma > 0$, $F \geq 0$, and $1 \geq \Lambda > 0$. We first consider the case $R = c$, which implies $r = c$. The optimisation problem now reduces to two variables λ and α, which is convex, and hence can be effectively solved using the KKT method, which gives the optimal contract identical to that in the proposition.

When $R > c$, we consider the gradients of all inequality constraints on three variables α, λ and r:

$$v_1 = \nabla(r - R) = (0, 0, 1);\ v_2 = \nabla(c - r) = (0, 0, -1);$$
$$v_3 = \nabla(c(1 + \alpha) - (f + r)(\lambda + 2\alpha)) = (c - 2(f + r), -(f + r), -2\alpha - \lambda);$$
$$v_4 = \nabla(-\alpha) = (-1, 0, 0); v_5 = \nabla(-\lambda) = (0, -1, 0);$$
$$v_6 = \nabla(\lambda - \Lambda) = (0, 1, 0);\ v_7 = \nabla(\lambda - (1 - \alpha)) = (1, 1, 0).$$

We now prove that there are no positive-linear dependencies among these vector, i.e., there exist no $a_i \geq 0$ with $i = \overline{1,7}$ not all zero, such that $\sum_{i=1}^{7} a_i v_i = \mathbf{0}$, in that $a_i > 0$ implies the i-th inequality is active. We first starts with $a_1 > 0$, which implies $r = R$, and that either $a_2 > 0$ or $a_3 > 0$ to form a linear dependency. However, $a_2 > 0$ implies $r = c < R$, which is contradicting, and thus we have $a_3 > 0$. However, the first component of v_3 is $c - 2(f + r) < 0$, therefore we need $a_7 > 0$ for linear dependency, giving $\lambda = 1 - \alpha$. If no other vectors are involved, then it must be that $c - 2(f + r) = -(f + r)$ (because $v_7 = (1, 1, 0)$), implying $c = f + r = f + R > c$, which is contradictory. Otherwise when $c - 2(f + r) < -(f + r)$, we need $a_5 > 0$, as it is the only remaining vector with the second component being negative, implying $\lambda = 0$ and thus $\alpha = 1$. Plugging this to the active third constraint would give $2c - 2(f + R) = 0$, which is not possible. This concludes that $a_1 = 0$ in all linear dependency. This also indicates $a_2 = 0$, as otherwise there is no vector (apart from v_1) with the third component being positive to form a linear dependency with v_2.

Meanwhile, $a_3 > 0$ implies the last component of v_3 is $-2\alpha - \lambda = 0$, or $\alpha = \lambda = 0$. However, we recall that $a_3 > 0$ implies $a_7 > 0$, which means that $\lambda = 1 - \alpha$, which is impossible when $\alpha = \lambda = 0$. This concludes that $a_3 = 0$. For if $a_4 > 0$ and $\alpha = 0$, then because the first component of v_4 is negative, we need $a_7 > 0$ and $\lambda = 1 - \alpha$, which likewise implies $a_5 > 0$ and $\lambda = 0$, which is again contradicting, thus $a_4 = 0$. Finally, a_5 and a_6 cannot be both positive, because that means $\Lambda = 0$. This concludes the proof that there is no positive-linear dependency in all possible contracts. □

Corollary 4.1. *If auditing is more expensive than the cost of honest computation ($\gamma \geq c$), the optimal contract only uses the redundancy scheme. When $\gamma \leq c/2$, either there is no redundancy scheme ($\alpha = 0$) or the whole auditing capacity is used ($\lambda^* = \Lambda$).*

The first part of the corollary is quite intuitive: when $\gamma > c$, any instance of outsourcing to a single agent and performing independent auditing can be replaced by the redundancy scheme (job duplication) and strictly lower the cost by $\gamma - c$.

Further Discussion. First, note that in our optimal two-agent contract, as long as $R \geq 2c$, there is no infeasible region: there is always a contract that makes $(\mathscr{H}, \mathscr{H})$ an equilibrium. Moreover, the payment to any of the agents is never more than the cost of honest computation. Figure 4.4a provides a pictorial representation of the proposition where $c/2 < \gamma < c$ and $\Lambda = 0.5$. When the enforceable fine is large, the redundancy scheme is preferable. This is despite the fact that the redundancy scheme is more expensive than auditing: it costs an extra c as opposed to $\gamma < c$. In other words, for high values of fine, the redundancy scheme is a more effective threat against cheating than independent auditing. When F is less than F_1, the independent auditing becomes the preferred method. For lower values of F, when the auditing capacity is all used up, the redundancy scheme is added to compensate for the low value of fine to maintain incentive compatibility. Figure 4.4b depicts the effect of auditing capacity, Λ, on the optimal contract where $c/2 < \gamma < c$. When $\Lambda = 0$, redundancy scheme is the only means to enforce honest computation. If furthermore no fine can be enforced ($F = 0$), then $\alpha = 1$: the job should be always duplicated. As Λ increases, there is a gradual transition from using redundancy scheme to independent auditing ($F < F_1$).

4.5.2 Global Optimality of Two-Agent Contracts

In developing the optimal contracts for two-agent case, we made a few critical assumptions: (a) the independent auditing is perfect; (b) the agents are non-colluding and non-communicating; (c) the range of intermediate steps is large enough that the probability of any two guessed results to be same, or the guessed result to be the correct

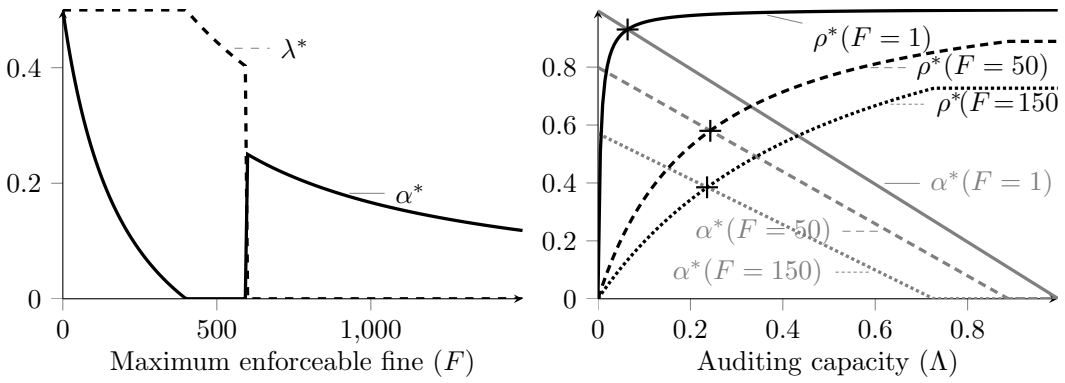


Figure 4.4: Optimal contract (where $c = 400, \gamma = 250$) w.r.t. (a) max. enforceable fine F ($\Lambda = 0.5$); and (b) auditing capacity Λ ($F_1 = 600$). Recall $\rho = \frac{\lambda}{1-\alpha}$ is the conditional probability of auditing given the job is assigned to a single agent.

result, is negligible; and (d) the agents are lazy but non-malicious. It turns out that these assumptions are sufficient to warrant global optimality of two-agent contracts among contracts that engage any number of agents in the following notion:

Proposition 4.6. *The contract that hires at most two agents and chooses its terms according to proposition 4.5, is globally optimal, that is, it achieves the least cost to the principal among all contracts that employ any number of agents and aim to make honest computation a Nash Equilibrium.*

Proof. Our approach in proving global optimality is to show that any other form of contract gives neither a stronger mean of punishment, nor less cost of operation to the principal. First, we provide an argument that if the optimal contract indeed assigns a task to more than one agent, then it does not benefit from hiring them in a "sequential" manner. Next, we prove that two-agent contract is the best solution among all "non-sequential" contracts by converting any given contract to a two-agent contract and improving the cost of the outsourcer.

Suppose that the optimal contract hired the agents sequentially for a given task. If in the first step, more than one agent is hired, then there are two possibilities: either (a) the returned results are the same, in which case, the computation is correct and there is no point in hiring any more agents and any subsequent steps; or (b) the returned results are different, which means that at least one of the agents has not computed the task correctly, in which case, all of the agents can be punished which includes the wrongdoer, and hence there is no need for any subsequent steps. If in the first step only one agent is assigned the task, there are again two possibilities: (a) the agent is audited, in which case the principal unequivocally knows whether cheating has

occurred and hence, there is no need for subsequent steps; or (b) the returned result is not audited. In the latter case, because agents never communicate, the agents that are hired in the next immediate step can be combined with the single agent hired in the first step as though they are all hired at the same time. Then, the argument for multiple agents in the first step can be applied to remove the need for any subsequent steps. Therefore, any optimal contract is either non-sequential or can be converted to a non-sequential one.

Now, let α_i be the probability that i agents are hired for $i \in \{2, \ldots, N\}$ where N is the maximum number of agents. Let λ_j be the (unconditional) probability that agent j is independently audited. Also let p_{ji} be the (conditional) probability that agent j receives the task if i agents are assigned. The expected cost of the contract to the principal is the following:

$$\mathcal{C} = r\sum_{i=2}^{N} i\alpha_i + r(1 - \sum_{i=1}^{N} \alpha_i) + \gamma \sum_{j=1}^{N} \lambda_j = r\sum_{i=2}^{N}(i-1)\alpha_i + r + \gamma \sum_{j=1}^{N} \lambda_j$$

Let φ_j be the probability that agent j receives the task. Then: $\varphi_j = \sum_{i=2}^{N} p_{ji}\alpha_i + p_{j1}(1 - \sum_{i=2}^{N} \alpha_i)$. The expected utility of agent j for honest computation given that it has received the message, and given that the rest of the agents are honest, is simply $r - c$. Now, let us define ψ_j to be the probability that agent j is rewarded given its strategy is to cheat. Given the honesty of all other agents, agent j is rewarded only if it is the only one that is assigned the task and it is not audited on. This gives: $\psi_j = p_{j1}(1 - \sum_{i=2}^{N} \alpha_i) - \lambda_j$. Therefore, the expected utility of agent j for cheating given it is assigned the task and all other agents are honest is $r\psi_j/\varphi_j - f(\varphi_j - \psi_j)/\varphi_j$. Hence, the incentive compatibility constraint, i.e., $u_j(\mathscr{H}, \overline{\mathscr{H}}) \geq u_j(\mathscr{C}, \overline{\mathscr{H}})$ with $\overline{\mathscr{H}}$ representing the honest strategy of other $N-1$ agents, becomes: $r \geq \dfrac{\varphi_j}{\varphi_j - \psi_j} c - f$. Taking into account the incentive compatibility of all agents, we have:

$$\frac{r+f}{c} \geq \max_j \frac{\sum_{i=2}^{N} p_{ji}\alpha_i + p_{j1}(1 - \sum_{i=2}^{N} \alpha_i)}{\sum_{i=2}^{N} p_{ji}\alpha_i + \lambda_j}$$

The participation constraint given that the honesty of all agents is established is simply $r - c \geq 0$. Thus, the optimal contract is given by the following optimisation:

$$\min_{\alpha_i, \lambda_j, p_{ji}, r, f} \mathcal{C} = r\sum_{i=2}^{N}(i-1)\alpha_i + r + \gamma \sum_{j=1}^{N} \lambda_j$$

$$\text{s.t.} \quad Nr \leq R,\ f \leq F,\ \lambda_j, p_{j1}, \alpha_i \geq 0,\ \lambda_j \leq (1 - \sum_{i=2}^{N} \alpha_i)p_{j1},\ \sum_{j=1}^{N} p_{ji} = 1\ \forall i,$$

$$\sum_{j=1}^{N}\lambda_j+\sum_{i=2}^{N}\alpha_i\leq 1,\sum_{j=1}^{N}\lambda_j\leq\Lambda,\ \frac{r+f}{c}\geq\max_j\frac{\sum_{i=2}^{N}p_{ji}\alpha_i+p_{j1}(1-\sum_{i=2}^{N}\alpha_i)}{\sum_{i=2}^{N}p_{ji}\alpha_i+\lambda_j},\ r-c\geq 0.$$

Now, suppose there is a claimed solution in which $\alpha_i>0$ for at least one $i\in\{3,\ldots,N\}$. In what follows we construct an alternative solution that improves the cost to the principal in which $\alpha_i=0$ for all $3\leq i\leq N$. Consider this alternative contract:

$$\hat{\alpha}_2=\sum_{i=2}^{N}\alpha_i,\ \hat{\alpha}_i=0\ \forall i\geq 3,\ \hat{\lambda}_j=\sum_{k=1}^{N}\lambda_k/N,\hat{p}_{ji}=1/N\ \forall i,j,\ \hat{r}=r,\ \hat{f}=f.$$

First, we show that given the feasibility of the claimed contract, this alternative contract is also feasible, and subsequently, establish the improvement in the achieved cost. The only non-trivial constraint to check for feasibility of the above contract is the incentive compatibility constraint:

$$\frac{r+f}{c}\geq\frac{\frac{1}{N}\sum_{i=2}^{N}\alpha_i+\frac{1}{N}(1-\sum_{i=2}^{N}\alpha_i)/N}{\frac{1}{N}\sum_{i=2}^{N}\alpha_i+\frac{1}{N}\sum_{j=1}^{N}\lambda_j}=\frac{1}{\sum_{i=2}^{N}\alpha_i+\sum_{j=1}^{N}\lambda_j}\tag{4.15}$$

From the feasibility of the claimed contract, we have:

$$\begin{aligned}&\frac{r+f}{c}\geq\max_j\frac{\sum_{i=2}^{N}p_{ji}\alpha_i+p_{j1}(1-\sum_{i=2}^{N}\alpha_i)}{\sum_{i=2}^{N}p_{ji}\alpha_i+\lambda_j}\\&\Rightarrow\frac{r+f}{c}\geq\frac{\sum_{j=1}^{N}\left(\sum_{i=2}^{N}p_{ji}\alpha_i+p_{j1}(1-\sum_{i=2}^{N}\alpha_i)\right)}{\sum_{j=1}^{N}\left(\sum_{i=2}^{N}p_{ji}\alpha_i+\lambda_j\right)}\\&=\frac{\left(\sum_{i=2}^{N}\sum_{j=1}^{N}p_{ji}\alpha_i+\sum_{j=1}^{N}p_{j1}(1-\sum_{i=2}^{N}\alpha_i)\right)}{\left(\sum_{i=2}^{N}\sum_{j=1}^{N}p_{ji}\alpha_i+\sum_{j=1}^{N}\lambda_j\right)}=\frac{\sum_{i=2}^{N}\alpha_i+(1-\sum_{i=2}^{N}\alpha_i)}{\sum_{i=2}^{N}\alpha_i+\sum_{j=1}^{N}\lambda_j}\end{aligned}$$

which gives (4.15). This establishes that the new solution is also feasible. In the first line of the above argument, we used the following simple lemma:

Lemma 4.1. *If we have* $a\geq\max_{j\in\mathcal{J}}\frac{b_j}{c_j}$ *where* $c_j>0$ *for all* $j\in\mathcal{J}$*, then* $a\geq\frac{\sum_{j\in\mathcal{J}}b_j}{\sum_{j\in\mathcal{J}}c_j}$.

Now: $\hat{\mathcal{C}}=\hat{r}\sum_{i=2}^{N}(i-1)\hat{\alpha}_i+\hat{r}+\gamma\hat{\lambda}=r\sum_{i=2}^{N}\alpha_i+r+\gamma\sum_{j=1}^{N}\lambda_j\leq r\sum_{i=2}^{N}(i-1)\alpha_i+r+\gamma\sum_{j=1}^{N}\lambda_j=\mathcal{C}$. □

The above proposition shows that our contract for two agents is not just a special case solution of multiple agents, but it is indeed the solution involving any number of agents. In other words, given the stipulated assumptions, there is no advantage ever in hiring more than two agents. Incidentally, we also show that the best contracts makes

the probability of any of the agents to be hired equal. This makes intuitive sense, as unequal probability of task assignment creates some "information" which the agents can potentially exploit to their benefit, and to the detriment of the principal.

4.6 Side-Channel (Information Leakage)

One of the important assumptions we made in developing our optimal hybrid contract was that the two agents do not communicate, and hence, upon receiving a task, an agent is not aware whether the same task is assigned to another agent or not. The principal uses this ambiguity in its favour to enhance the threat of auditing through redundancy. However, if agents somehow gain access to this information, the threat loses its efficacy. Specifically, if redundancy scheme is used, an agent can selectively be honest if it finds out that the task is outsourced to another agent (hence the name *side channel*), and be lax when it knows it is the only recipient of the task. If the principal supposes such "information leakage", then the contract optimisation problem must be modified. Note that the agents now have two distinct information states: one in which they are the only assignee and another in which, both of them have received the task. We refer to them as *lone recipient* and *redundancy* information states, respectively. The privacy of information states implies incompleteness of information, and thus the game Γ_o played between the agents is no longer a simple "one-shot" game, but becomes a Bayesian game. Particularly, Γ_o is of the form $\langle A, \Omega, \langle D_i, T_i, C_i, \tau_i, p_i, u_i \rangle_{i \in A} \rangle$, with $\Omega = \{1, 2, both\}$, representing employment of agent 1, 2, or both, respectively. For each $(s_1, s_2) = \omega \in \Omega$, $\tau_1(1) = L$, $\tau_1(2) = 0$, $\tau_1(both) = R$, implying the lone recipient state, unemployment, or redundancy state. Also, agents' view on Ω is public, i.e., $p_1(s_1, s_2) = p_2(s_1, s_2) = p(s_1, s_2)$ for a probability distribution p induced by the choice of α in the chosen contract offer $o = (r, f, \alpha, \lambda)$. The new (ex-post) utility function effectively becomes, for agents' strategies $d_1, d_2 : \{L, 0, R\} \to [0, 1]$:

$$
\begin{aligned}
u_i(\omega = i, d_1, d_2) &= (1 - (1 - d_i(L))\rho)r - (1 - d_i(L))\rho f - c d_i(L) \\
u_i(\omega = -i, d_1, d_2) &= 0 \\
u_i(\omega = both, d_1, d_2) &= d_i(R) d_{-i}(R) r - (1 - d_i(R) d_{-i}(R)) f - c d_i(R)
\end{aligned}
$$

The constraints for the general optimisation problem in (4.1) must then be modified, so that $(\mathscr{H}, \mathscr{H})$ is a Bayesian Nash equilibrium, where $\mathscr{H}(\cdot) = 1$. Due to the fact that τ_i is one-to-one, and that u_i is linear in d_i, such requirement of equilibrium translates to $u_i(\cdot, \mathscr{H}, \mathscr{H}) \geq u_i(\cdot, \mathscr{C}, \mathscr{H})$ and $u_i(\cdot, \mathscr{H}, \mathscr{H}) \geq u_i(\cdot, \perp, \mathscr{H}) = 0$, where $\mathscr{C}(\cdot) = 0$. The new incentive compatibility constraint (preferring honest computation over cheating)

for agent i in the lone recipient information state ($\omega = i$) is thus:

$$r - c \geq r(1-\rho) - f\rho \Leftrightarrow r\lambda \geq c(1-\alpha) - f\lambda \tag{4.16}$$

For the redundancy information state, the incentive compatibility is: $r - c \geq -f$, because if the agent cheats, the results will be different and the agent will definitely be punished.[15] This constraint is redundant, because the participation constraint is still $r - c \geq 0$, which also implies $r - c \geq -f$. Here, we assume $R \geq 2c$. This will allow us to ignore the budget constraint, which is: $r \leq R$ if $\alpha = 0$, and $2r \leq R$ if $\alpha > 0$. This is because the optimal contract turns out to choose $r^* = c$ and hence, for $R \geq 2c$, the budget constraint is automatically satisfied. [16] Hence, the new optimisation problem is the following:

$$\min_{r,f,\alpha,\lambda} r(1+\alpha) + \gamma\lambda \text{ subject to:} \tag{4.17a}$$

$$f \leq F,\ 0 \leq \lambda \leq \Lambda,\ \lambda \leq 1-\alpha,\ \alpha \geq 0,\ r \geq c,\ r\lambda + f\lambda \geq c(1-\alpha). \tag{4.17b}$$

The solution is given as the following proposition:

Proposition 4.7. *The optimal two-agent contract with information leakage, i.e., where the agents have access to the information of whether the same task is outsourced to the other agent or not, enforces honesty in that makes $(\mathscr{H}, \mathscr{H})$ a Nash equilibrium sets $f^* = F$, $r^* = c$, and:*

$$\gamma \geq \frac{c}{\Lambda}: \begin{cases} F \geq [\gamma - c]^+ : & \lambda^* = \dfrac{c}{c+F},\ \alpha^* = 0,\ \mathcal{C}^* = c + \dfrac{\gamma c}{c+F} \\ F < [\gamma - c]^+ : & \lambda^* = 0,\ \alpha^* = 1,\ \mathcal{C}^* = 2c \end{cases}$$

$$\gamma < \frac{c}{\Lambda}: \begin{cases} F \geq [c/\Lambda - c]^+ : & \lambda^* = \dfrac{c}{c+F},\ \alpha^* = 0,\ \mathcal{C}^* = c + \dfrac{\gamma c}{c+F} \\ [\gamma - c]^+ \leq F < [c/\Lambda - c]^+ : & \lambda^* = \Lambda,\ \alpha^* = 1 - \Lambda(1 + \dfrac{F}{c}),\ \mathcal{C}^* = c(2 - \Lambda(1 + \dfrac{F}{c})) + \gamma\Lambda \\ F < [\gamma - c]^+ : & \lambda^* = 0,\ \alpha^* = 1,\ \mathcal{C}^* = 2c \end{cases}$$

Proof. We first argue that we can safely assume that the fine is at its maximum value, i.e., $f^* = F$: the principal can manipulate r, λ, α or f in order to enforce the incentive compatibility constraint in the lone recipient information state. Among these variables, only increasing the fine is costless to the principal. Moreover, the only two constraints that f appears in is $f \leq F$ and the incentive compatibility constraint. Hence, any

[15] Note that even when the agents know that the redundancy scheme is being used, unless they coordinate their reported results, guessed results will be the same only with negligible probability. We will consider the case of collusion in the next section.

[16] Recall that for $R < 2c$, the outsourcer can never assign more than one agent with any positive probability, and hence, the issue of information leakage is irrelevant.

optimal contract can be transformed to one in which $f = F$, keeping all other parameters fixed. We use the Karush-Kuhn-Tucker (KKT) conditions [11] to solve the above non-linear (non-convex) programming. The non-convexity arises due to the incentive compatibility constraint (the last constraint in (4.17b)). Note that our cost and constraint functions are all continuously differentiable. We first use the Mangasarian–Fromovitz constraint qualification (MFCQ) to establish that any minimum must satisfy the KKT conditions, i.e., KKT are necessary conditions of optimality. In the absence of equality constraints, the MFCQ condition means that the gradients of the active inequality constraints are positive-linearly independent at optimum points.

As we mentioned before, we assume $R \geq 2c$, since otherwise, never more than one agent can be hired. It will turn out that the optimal contract will always choose $r^* = c$, hence the budget constraint of $r \leq R$ for $\alpha = 0$ and $r \leq 2R$ for $\alpha > 0$ is automatically satisfied. The remaining inequalities (written in standard form) are $-\lambda \leq 0$, $\lambda - \Lambda \leq 0$, $\lambda + \alpha - 1 \leq 0$, $-\alpha \leq 0$, $c - r \leq 0$, $c(1-\alpha) - F\lambda - r\lambda \leq 0$. The gradients of these inequality constraints with the order of variables as (λ, α, r) are: $(-1, 0, 0)$, $(1, 0, 0)$, $(1, 1, 0)$, $(0, -1, 0)$, $(0, 0, -1)$ and $(-F - r, -c, -\lambda)$. We will consider the cases of $\alpha = 1$ and $\alpha < 1$ separately.

If $\alpha = 1$, we must have $\lambda = 0$, which means the only possible active inequalities are $-\lambda \leq 0$, $\lambda + \alpha \leq 1$ and $c - r \leq 0$, with gradients $(-1, 0, 0)$, $(1, 1, 0)$ and $(0, 0, -1)$. These gradients are clearly linearly independent and the MFCQ condition holds. If $\alpha < 1$, from the last constraint, we must have $\lambda > 0$. Now consider two cases: $\Lambda = 1$ or $\Lambda < 1$. When $\Lambda = 1$, then the constraint of $\lambda \leq \Lambda$ is implied by $\lambda + \alpha \leq 1$ and $\alpha \geq 0$, and can be removed. Now, if $\alpha = 0$, then the last inequality $c(1-\alpha) - F\lambda - r\lambda$ is implied by $c - r \geq 0$ and hence can be removed. The standing inequalities will therefore be $\lambda + \alpha \leq 1$, $c - r \leq 0$, along with the active inequality of $-\alpha \leq 0$. The gradients of these inequalities are respectively $(1, 1, 0)$, $(0, 0, -1)$ and $(0, -1, 0)$, and are clearly linearly independent. If, on the other hand, $0 < \alpha < 1$, then the standing constraints are: $\lambda + \alpha - 1 \leq 0$, $c - r \leq 0$ and $c(1-\alpha) - F\lambda - r\lambda \leq 0$, with the gradients: $(1, 1, 0)$, $(0, 0, -1)$ and $(-F - r, -c, -\lambda)$. The last three vectors are linearly independent because all the elements of the last vector are non-zero given $\lambda > 0$. When $\Lambda < 1$, first suppose $\alpha = 0$. Then, the constraint of $\lambda + \alpha \leq 1$ is inferred from $\lambda \leq \Lambda$, and hence can be removed. The standing constraints are $\lambda - \Lambda \leq 0$, $c - r \leq 0$, $c(1-\alpha) - F\lambda - r\lambda \leq 0$, along with the active constraint of $-\alpha < 0$. The gradients of these constraints are $(1, 0, 0)$, $(0, 0, -1)$, $(-F - r, -c, -\lambda)$ and $(0, -1, 0)$. Note that except for the singleton point of $F = c(1/\Lambda - 1)$, never all four of these constraints are active.[17] Now note that any three of the gradients are linearly independent given $\lambda > 0$. Finally, when

[17] The optimal contract for such a point can be derived using a continuity argument.

$\Lambda < 1$ and $0 < \alpha < 1$, the standing constraints are $\lambda - \Lambda \leq 0$, $\lambda + \alpha - 1 \leq 0$, $c - r \leq 0$ and $c(1+\alpha) - F\lambda - r\lambda \leq 0$, with gradients $(1,0,0)$, $(1,1,0)$, $(0,0,-1)$ and $(-F-r, -c, -\lambda)$, respectively. As before, the only case that all four of these constraints can be active is the single point of $F = c - r$. For all other values of F, at most three of these constraints are active, whose gradients are linearly independent given $\lambda > 0$. In summary, the MFCQ normality condition holds.

To systematically obtain the KKT conditions, we introduce the dual multipliers μ_1 to μ_6, and transform the problem in (4.17) as follows:

$$\min_{r,\alpha,\lambda,\mu_i} \bar{\mathcal{C}} = r(1+\alpha) + \gamma\lambda - \mu_1\lambda + \mu_2(\lambda - \Lambda) + \mu_3(\lambda + \alpha - 1) \\ - \mu_4\alpha + \mu_5(c - r) + \mu_6\left(c(1-\alpha) - F\lambda - r\lambda\right) \quad (4.18)$$

subject to:

$$\text{primal feasibility: } 0 \leq \lambda \leq \Lambda,\ \lambda \leq 1 - \alpha, \\ \alpha \geq 0,\ r \geq c,\ r\lambda + F\lambda \geq c(1-\alpha) \quad (4.19a)$$

$$\text{dual feasibility: } \mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6 \geq 0, \quad (4.19b)$$

$$\text{complementary slackness: } \mu_1\lambda = 0,\ \mu_2(\lambda - \Lambda) = 0, \quad (4.19c)$$

$$\mu_3(\lambda + \alpha - 1) = 0, \mu_4\alpha = 0, \quad (4.19d)$$

$$\mu_5(c - r) = 0, \mu_6\left(c(1-\alpha) - F\lambda - r\lambda\right) = 0. \quad (4.19e)$$

The first order conditions of optimality are:

$$\frac{\partial \bar{\mathcal{C}}}{\partial \lambda} = 0 \Leftrightarrow \gamma - \mu_1 + \mu_2 + \mu_3 - \mu_6(F + r) = 0, \quad (4.20)$$

$$\frac{\partial \bar{\mathcal{C}}}{\partial \alpha} = 0 \Leftrightarrow r + \mu_3 - \mu_4 - c\mu_6 = 0, \quad (4.21)$$

$$\frac{\partial \bar{\mathcal{C}}}{\partial r} = 0 \Leftrightarrow (1+\alpha) - \mu_5 - \lambda\mu_6 = 0. \quad (4.22)$$

The full solution is now derived as in the statement of the proposition by straightforward investigation of the conditions (4.19) through (4.22). □

Discussion Firstly, note that the cost of the above contract is clearly higher than that of the contract with no information leakage, but lower than the cost of single-agent contract. The latter is because the single-agent contract in (4.4) is a feasible solution of the above optimisation by setting $\alpha = 0$. In fact, the cost of the contract is capped at $2c$: when both agents are hired at all times (i.e., with probability one), the incentive

compatibility constraint and participation constraint are clearly satisfied with $r = c$, giving the contract cost of $2c$. This makes the redundancy scheme still an appealing option specially when the cost of auditing is high or there is little (or zero) capacity for auditing. Secondly, note that for $\gamma < c$, the redundancy scheme is never used for any value of maximum enforceable fine. When the cost of independent auditing γ is higher than the cost of honest computation c, if the enforceable fine is below the threshold of $\gamma - c$, it is best to only use the redundancy scheme, but it has to be done with certainty, i.e., $\alpha^* = 1$. Note that with information leakage, never probabilistic usage of redundancy scheme *alone* is optimal, because the agents will choose to be lazy when they *know* they are not audited. Also like the two-agent contract with no information leakage (but unlike the single-agent), the optimal reward r^* is never higher than the cost of honest computation c, irrespective of the value of the maximum enforceable fine. Moreover, we observe that for large values of enforceable fine, in contrast to the no information leakage case, it is the independent auditing mechanism that is now the preferred method.

4.7 Colluding Agents

Suppose the agents not only know the state of the other agents with respect to the task assignment, but they can also coordinate their response to report the same guessed result. This can save them from the cost of honest computation and at the same time, go undetected. Hence, unlike before, returned results from multiple agents that are the same may not be correct. The principal can audit the returned results (through re-computation of the task) when they are the same. Consider two agents. As in the information leakage setting, each agent has two distinct information state: being the sole recipient, and being one of the two recipients.

The Bayesian game Γ_o for collusion is similar to the case of information leakage, with the exception of players' utilities when $\omega = both$. This is because when both agents cheat, they would be rewarded instead of being punished, and hence

$$u_i(\omega = both, d_1, d_2) = (1 - (d_i + d_{-i} - 2d_i d_{-i}))r - (d_i + d_{-i} - 2d_i d_{-i})f - cd_i \quad (4.23)$$

Note that similar to the case of information leakage, the principal still has to enforce honesty when an agent is in its *lone recipient* information state. This implies ((4.16)):

$$r \geq \frac{c}{\rho} - f \quad (4.24)$$

In this setting it might still be possible to make $(\mathscr{H}, \mathscr{H})$ an equilibrium as in previous

settings. However, it is also possible for $(\mathscr{C},\mathscr{C})$ to be an equilibrium, in which case the agents would receive utility $u_i(both,\mathscr{C},\mathscr{C}) = r$ which is better than $u_i(both,\mathscr{H},\mathscr{H}) = r - c$, as they manage to collude and cheat the principal whilst saving the cost of computation. Thus the agents would prefer $(\mathscr{C},\mathscr{C})$ over $(\mathscr{H},\mathscr{H})$, and thus we might fail to guarantee fully honest computation. One way to dissuade the agents from colluding is to make collusion a less attractive equilibrium than honesty. To do so, we add another threat: the returned results from the two agents are audited by the principal with probability ν, (even) when they are the same. Note that the value of ν enters the cost of the principal even if honest computation is indeed enforced. This is because when honest computation is enforced, the returned results are the same too. Specifically, there will be an additional term of $\gamma\alpha\nu$ to the principal's cost. With this modification, agent i's utility for redundancy changes from (4.23) to

$$\begin{aligned} u_i(\omega = both, d_1, d_2) = {} & d_i d_{-i} r + (1-d_i)(1-d_{-i})(1-\nu) r \\ & - (d_i(1-d_{-i}) + (1-d_i)d_{-i}) f - (1-d_i)(1-d_{-i})\nu f \end{aligned}$$

which implies $u_i(both,\mathscr{C},\mathscr{C}) = r(1-\nu) - f\nu$. Therefore, to make honesty a more attractive equilibrium than collusion, in the redundancy scheme information state, we must have:

$$r - c \geq r(1-\nu) - f\nu \Leftrightarrow r \geq \frac{c}{\nu} - f \tag{4.25}$$

Hence, the corresponding optimal contract is given by the following optimisation:

$$\min_{r,f,\alpha,\lambda,\nu} r(1+\alpha) + \gamma\lambda + \gamma\alpha\nu, \quad \text{s. t. :}\ r \leq R,\ f \leq F,\ 0 \leq \rho \leq 1,\ 0 \leq \nu \leq 1 \tag{4.26a}$$

$$\text{and:}\quad \rho(1-\alpha) + \alpha\nu \leq \Lambda,\ \alpha \geq 0,\ r \geq c,\ r \geq \frac{c}{\rho} - f,\ r \geq \frac{c}{\nu} - f. \tag{4.26b}$$

We have the following proposition:

Proposition 4.8. *The optimal contract that enforces honesty in lone information state and makes collusion a less attractive equilibrium than honest computation in the redundancy information state sets $\alpha^* = 0$, i.e., never uses the redundancy scheme at all. The rest of the parameters of the contract are also according to the optimal contract for a single agent provided in* (4.6).

The result can be derived directly by examining the KKT conditions of the optimisation problem in (4.26) after establishing that KKT conditions are indeed applicable. However, we provide a simpler proof that delivers more intuition.

Proof. Consider a claimed optimal contract that selects an $\alpha > 0$. We will construct an alternative feasible contract that employs only one agent ($\alpha_{\text{alt}} = 0$) and strictly improves the cost, hence reaching a contradiction.

The claimed contract has to satisfy inequalities (4.24) and (4.25) to be feasible. Now consider an alternative contract alt that only selects one agent and audits it with probability $\lambda_{\text{alt}} = \rho(1-\alpha) + \alpha\nu$. The values of the reward and fine are the same. Fist we examine the change in the contract cost: $C_{\text{alt}} - C = [r + \gamma\left(\rho(1-\alpha) + \nu\alpha\right)] - [r(1+\alpha) + \gamma\rho(1-\alpha) + \gamma\nu\alpha] = \alpha r$, which is strictly positive based on the assumption that $\alpha > 0$. Now if we show that this alternative contract is feasible, then we have reached the contradiction we are after.

The only non-trivial constraint that we need to verify to establish the feasibility of the alternative contract is the incentive compatibility: we must have:

$$r - c \geq r(1-\lambda_{\text{alt}}) - f\lambda_{\text{alt}} \Rightarrow r \geq \frac{c}{\lambda_{\text{alt}}} - f \quad \text{that is:} \quad r \geq \frac{c}{\rho(1-\alpha) + \nu\alpha} - f$$

The last inequality can be inferred from (4.24) and (4.25) and the fact that for any $\alpha \geq 0$, we have: $\rho(1-\alpha) + \nu\alpha \leq \min(\rho, \nu)$. This completes the proof. □

Intuitively, whenever the two agents are to be assigned, the principal can save the reward to the second agent by assigning the task to only one of them. The principal will audit the only agent as it would have audited the two agents. This works since two colluding agents (so far) act as though a single agent anyway.

The above proposition is a negative result: the benefits of redundancy scheme seem to be all lost if the principal suspects collusion between the agents. However, in what follows, we introduce two schemes based on the idea of offering "bounties" that, at least partially, save the redundancy scheme. These bounty schemes better utilise the incentive of the agents against each other, creating a prisoner's dilemma-like situation to undermine collusion. That is, instead of trying to make collusion a less attractive equilibrium, which we observed is futile in Proposition 4.8, these schemes make collusion a non-equilibrium. The idea is as follows: when the returned results are different, with probability β the principal can audit the task and reward the bounty in such cases to the agent with the correct result (if any). The value of the bounty should be the largest credible promise, i.e., R. The difference between the two schemes is how they treat the unaudited cases when the returned results are different:

- *Bounty scheme one*: when the results are different and auditing does not occur, both agents are punished at f, which implies,

  $$u_i(\omega = both, d_1, d_2) = d_i d_{-i} r + (1-d_i)(1-d_{-i})(1-\nu)r$$

$$+ (d_i(1-d_{-i}) + (1-d_i)d_{-i})\beta R$$
$$- (d_i(1-d_{-i}) + (1-d_i)d_{-i})(1-\beta)f - (1-d_i)(1-d_{-i})\nu f$$

- *Bounty scheme two*: when the returned results are different and the task is not audited, both agents are rewarded at r, i.e.,

$$u_i(\omega = both, d_1, d_2) = d_i d_{-i} r + (1-d_i)(1-d_{-i})(1-\nu)r$$
$$+ (d_i(1-d_{-i}) + (1-d_i)d_{-i})\beta R$$
$$- (d_i(1-d_{-i}) + (1-d_i)d_{-i})(1-\beta)r - (1-d_i)(1-d_{-i})\nu f$$

A nice feature of both schemes is that, if they indeed succeed to enforce honesty, the bounties will never in fact be paid: All that is necessary is the credible promise of the bounties. In what follows we analyse these two schemes. For both schemes, we have:

$$u_i(both, \mathscr{C}, \mathscr{C}) = r(1-\nu) - f\nu, \qquad u_i(both, \mathscr{H}, \mathscr{H}) = r - c$$

In bounty scheme one:

$$u_i(both, \mathscr{H}, \mathscr{C}) = -c + R\beta - f(1-\beta) \qquad u_i(both, \mathscr{C}, \mathscr{H}) = -f$$

In bounty scheme two:

$$u_i(both, \mathscr{H}, \mathscr{C}) = -c + R\beta + r(1-\beta), \qquad u_i(both, \mathscr{C}, \mathscr{H}) = r(1-\beta) - f\beta$$

Making $(\mathscr{H}, \mathscr{H})$ an equilibrium is automatic in scheme one: $r - c \geq -f$ for any $f \geq 0$, following the participation constraint $r - c \geq 0$. Making $(\mathscr{C}, \mathscr{C})$ a non-equilibrium requires the following:

$$-c + R\beta - f(1-\beta) \geq r(1-\nu) - f\nu \tag{4.27}$$

Similarly, making $(\mathscr{C}, \mathscr{C})$ a non-equilibrium for scheme two requires:

$$-c + R\beta + r(1-\beta) \geq r(1-\nu) - f\nu. \tag{4.28}$$

Moreover, to have $(\mathscr{H}, \mathscr{H})$ an equilibrium in scheme two, one must ensure:

$$r - c \geq r(1-\beta) - f\beta \tag{4.29}$$

In both cases, the value of β does not directly enter the cost of the contract to the principal if honesty is indeed enforced. Hence, the principal can choose the maximum possible value. In order to make it credible, the principal must have enough auditing capacity. Specifically, $\lambda+\alpha\beta \leq \Lambda$. Hence the maximum value of β is given as $(\Lambda-\lambda)/\alpha$. Replacing in (4.27) we obtain the following extra (incentive compatibility) constraint for scheme one:

$$(R+f)(\Lambda-\lambda)+(r+f)\alpha\nu-(c+r+f)\alpha \geq 0. \tag{4.30}$$

Similarly, replacing $\beta=(\Lambda-\lambda)/\alpha$ in (4.28) and (4.29) yields the following for scheme two:

$$(r+f)(\Lambda-\lambda)-\alpha c \geq 0, \qquad (R-r)(\Lambda-\lambda)+\nu(r+f)\alpha-c\alpha \geq 0 \tag{4.31}$$

Hence, the contract optimisation for bounty schemes one and two are the same as in (4.26) except that the last constraint in (4.26b), i.e., $r \geq c/\nu - f$, is replaced with (4.30) and (4.31), respectively. It turns out, however, that these two innocuous-looking optimisation problems do not lend easily to closed-form solutions.

In what follows we obtain partial solutions of these optimisations, which provide insight on the applicability of redundancy scheme in the presence of collusion. First, note that for any given set of parameters c, F, R, Λ, the best contract for the information leakage setting yields a better cost than any feasible contract in the collusion scenario (both bounty schemes). This is because, compared to (4.17), the optimisation problem of finding the best contract for bounty schemes one and two each have: (A) an additional non-negative term in the cost: $\gamma\alpha\nu$; and (B) an *extra* incentive compatibility constraint, (4.30) in scheme one and (4.31) in scheme two. Therefore, in particular, if a solution of the optimisation in (4.17) is a feasible solution for the optimisation of schemes one and two, then it is also optimal for them as well. This happens for example when the optimal information leakage contract chooses $\alpha = 0$, as then, the extra incentive compatibility constraint in (4.30) and (4.31) are trivially satisfied. Hence, in the light of Proposition 4.7, we have the following result:

Corollary 4.2. *For both schemes one and two, the optimal contract chooses* $\alpha^* = 0$ *for* $F \geq [\max(\gamma, c/\Lambda) - c]^+$. *The rest of the parameters for such cases are* $f^* = F$, $r^* = c$ *and* $\lambda^* = c/(c+F)$.

The corollary shows that for large values of the enforceable fine F, assigning a single agent is the preferred method of outsourcing. However, the corollary leaves out the question of whether redundancy scheme is ever the preferred method in the

presence of collusion with the introduction of the bounty schemes. The next result provides a positive answer. In particular, we derive sufficient conditions under which, the redundancy scheme is the preferred method even in the presence of collusion:

Corollary 4.3. *In bounty scheme two, for $F < [\gamma - c]^+$, if $\Lambda \geq c/\min(c+F, R-c)$, the optimal contract chooses redundancy $\alpha^* = 1$. The rest of the parameters for such a case are: $r^* = c$, $\lambda^* = \nu^* = 0$, $f^* = F$.*

The corollary is intuitive: the auditing capacity should be large enough to make the promise of checking for bounty when the results are different a credible one.

Proof. The corollary follows from a similar logic as in the previous corollary: we will find cases that the optimal solution of the information leakage contract optimisation in (4.17) are feasible solutions of the optimisation problem for scheme two. An alternative to $\alpha = 0$ is $\alpha = 1$: if $\nu = 0$ is a feasible choice for scheme two with the parameters that make $\alpha = 1$ an optimal solution for the information leakage setting, then the corresponding contract is optimal for scheme two. From Proposition 4.7, $\alpha^* = 1$ when $F < [\gamma - c]^+$. The rest of the parameters are $\lambda^* = 0$, $r^* = c$ and $f^* = F$. We should investigate whether these parameters and $\nu = 0$ satisfy (4.31), which becomes: $(c+F)\Lambda \geq c$ & $(R-c)\Lambda \geq c \Leftrightarrow \Lambda \geq c/\min(c+F, R-c)$, hence the corollary. □

The following corollary provides a sufficient condition for scheme one to use the redundancy scheme.

Corollary 4.4. *In bounty scheme one, for $F < [\gamma - c]^+$, if $\Lambda \geq 2c/R$, the optimal contract chooses redundancy $\alpha^* = 1$. The rest of the parameters for such a case are: $r^* = c$, $\lambda^* = \nu^* = 0$, and notably $f^* = 0$.*

Proof. The proof is similar to that of Corollary 4.3, with the following exception: Note from (4.30) that f plays a double edge sword role, and it is no more a priori clear that maximum fine is the best option. In fact for this case it turns out to be exactly the opposite. From Proposition 4.7, for $F \leq [\gamma - c]^+$, optimal contract is given by $r^* = c$, $\lambda^* = 0$, $\alpha^* = 1$ and $f^* = F$. However, the value of $f \geq 0$ for this region does not affect the cost and feasibility of the contract, and hence, any $f \geq 0$ is in fact also optimal. Replacing these parameters with a general f and along with $\nu = 0$ in (4.30), we obtain: $(R+f)\Lambda \geq 2c+f$. Hence a sufficient condition for feasibility (and hence optimality) is $(R+f)\Lambda \geq 2c+f$. The value of f is arbitrarily, the best result is obtained for $f = 0$, that is $\Lambda \geq 2c/R$. □

4.8 Contract Implementation

For completeness of the solutions, in this section we discuss notable technical concerns on the implementation of our contracts.

4.8.1 Intermediate Steps and Hash Functions

As we discussed in Section 4.3.1, the use of intermediate steps as part of the output would prevent trivial/clever guessing. However, the data representing intermediate steps could be large and thus cumbersome for transmission. [15] proposes the use of cryptographic hash as a sufficient representation of intermediate steps: Instead of sending a large amount of data detailing these steps, the agent can only send the cryptographic hash of such data. On receiving the agent's hash h_A, the principal repeats the computation, and computes its own hash h_P from the intermediate steps, then verifies that $h_A = h_P$.

Informally, the use of hash function is considered secure if it is unlikely that the agent can come up with the correct hash without knowing the correct intermediate steps. The authors in [15] require such hash function to be a "random oracle", i.e., a function mapping in which each output is chosen uniformly randomly regardless of the input. While this is a sufficient condition, the notion of random oracle is rather impractical, and also an overkill. Indeed, we argue that for this purpose of hash checking, it is necessary and sufficient that the hash function is "collision resistant", that is, it should be difficult to find two different messages with the same hash.

Lastly, note that the process of hashing the intermediate steps may itself carry a considerable cost. For instance, if the computation task is to hash a large string, then the cost of hashing the intermediate steps (if the same hash function is used) would be at least as much as computation cost. Therefore, either the cost of hashing intermediate steps must be negligible compared to that of the original computation task, or it must enter the contract model.

4.8.2 Enforcing The Principal's Auditing

With regards to legal enforcement of the contract, it is necessary that behaviours of contract participants are observable and verifiable. Actions such as "assigning a job" or "paying a reward" are of this type. However, the principal's action of "auditing a job" is not necessarily observable to the computing agent(s)/contractor(s), as it might be carried out offline by the principal. It is critical to ensure that the principal really performs auditing, for two reasons. Firstly, the principal must establish to the agents its commitment to auditing so as to make the threats credible. Secondly, the agent

needs an assurance that the principal cannot cheat and thus take away some of its benefits (in two-level rewarding). In this subsection we discuss a simple method for an agent to verify whether the principal has indeed properly performed an audit, where the auditing method is a simple re-computation. Of course such verification is only useful if it is a credible threat that the principal will be punished if it claims to have made an audit but is found otherwise. In this case we assume that the contract can be legally enforced by an authority (e.g., a court), and thus punishment on the principal's cheating/lying is guaranteed if there is enough evidence for the accusation.

Our audit mechanism relies on the use of a non-malleable commitment scheme $(\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open})$. Assume existence of a common reference string $\mathsf{CK} \leftarrow \mathsf{Setup}(k)$ for some security parameter $k > 0$. In its operation, the agent performs the computation to get a result r_A, then produces $(c_A, d_A) \leftarrow \mathsf{Commit}_{\mathsf{CK}}(r_A || \text{“agent”})$. The principal also perform the same operation, except that if it is meant not to audit, then it produces a random message of the same length. Whether or not this message is the result, we denote it as r_P, along with $(c_P, d_P) \leftarrow \mathsf{Commit}_{\mathsf{CK}}(r_P || \text{“principal”})$. The principal and the agent then exchange their commitment values c_P and c_A, respectively, followed by the exchange of d_P and d_A. The principal and the agent then open the commitments and learn the corresponding messages, say m_A and m_P, respectively. It is not difficult to see that this mechanism has the following properties:

- Honest auditing: by checking if $m_P = r_A || \text{“principal”}$, an honest agent can verify whether the principal has properly audited, since the use of intermediate steps as part of the output in the previous subsection guarantees that the principal cannot efficiently produce the correct output without honestly executing the whole computation. The principal also does not receive any useful information from the agent in creating m_P due to the hiding property of commitment, and it cannot produce a message m_P related to m_A due to the non-malleability of commitment, and that m_P cannot be the same as m_A (due to the additional suffix).

- Honest computation: Due to the binding property of commitment, the agent receives negligible advantage in producing a correct computation result from the execution of this mechanism. Due to the hiding property of commitment, it also cannot tell if the principal is meant to audit or not at the time of committing the computation result.

Using the above mechanism, the principal has negligible advantage in cheating during auditing, and thus for simplicity we assume from now on that the principal's action of “auditing a job” is securely observable and verifiable from the agent's view.

4.8.3 Enforcing Probabilistic Behaviours

Another problem with unobservable strategies is the principal's probabilistic behaviours. This becomes clear if we recall our Stackelberg game notion presented in Definition 4.1. Indeed, in the second stage of the game, the agents (followers) are informed of the contract offer, after which they must decide their course of action. While informing the offer to the agents can trivially be done, the realistic problem is on how to ensure that the principal sticks to its chosen strategy, i.e., whether it correctly follows the offer after the agents' acceptance. Indeed, an offer can be considered unobservable, as it might contain probabilistic behaviours, e.g., "employing two agents with probability α", are usually unverifiable. Our contracts unfortunately rely on these probabilistic actions of the principal as explicitly stated in the terms and policies for auditing, task duplication and/or rewarding (the latter in two-level reward contracts of subsection 4.4.3). Without an appropriate security measure, this is usually not possible, e.g., the fact that the principal does not audit tells little about whether its auditing probability is indeed $\lambda = 0.3$ or $\lambda = 0.6$. This important implementation issue has not been discussed in previous works.

Usually this could be achieved cryptographically using multiparty computation (MPC) [95], in which a sampling function on the principal's behaviour is accurately and securely computed among the contract participants. However, MPC assumes pairwise secure communication among participants, which in this case implies a need for direct communication between the agents. This poses a potential threat to our model: if agents can freely communicate, they may as well collude and give identically incorrect result, thus fooling the principal. Therefore we seek a mechanism that requires no agent-to-agent communication, in that it is easy for the agents to catch the principal cheating whenever the principal benefits from doing so.

In order to provably design such mechanism, we need to formalise the underlying security requirements, which capture concepts such as "behaviour", "deviation", "cheating", "catch". Informally, the behaviour of the principal is characterised by how it plans to act (and will eventually do so) in implementing the contract. A *plan* of actions for the principal essentially captures the its deterministic choices for all possible decision-making situations which might arise while executing the contract. An example of such plan could be: give the task to both agents; if the result coming back is the same, then reward both. Another example is: give the task to agent 1, then audit on return. For convenience we denote the set of all possible plans as Ω, which also contains an element $\perp$ representing an invalid plan. The principal P is supposed to pick a plan $\omega \in \Omega$ according to a contract-specific probability distribution $\Delta(\Omega)$, but the agents do not know if P actually follows this distribution, or a different one to

its eventual benefit. As a result, we decide to let such a plan be picked by the agents instead of the principal.

In order to pick a plan, the agents are required to communicate with each other, with their messages routed through the principal. This requires a well-specified communication protocol among the agents. Recall that our goal is to enforce the principal's behaviour to the contract terms. Essentially, the only way to ensure this is to employ a mechanism that detects the principal's cheating, as well as a viable option to punish the principal (e.g., by a court) accordingly. This is one key requirement that our protocol must satisfy. In addition, our communication protocol must ensure that it facilitates the nature of the contract, that is, it should accurately emulate the actual action of probabilistic sampling of behaviour by the principal itself.

More specifically, if the agents execute the protocol honestly, and that the principal does not tamper with the protocol messages, then the plan picked at the end of the execution must have probability measure according to $\Delta(\Omega)$. We name this requirement *correctness*. Also, if the principal is picking the plan on its own, the agents should not know about this, at least until after running the computation task. By emulating such event, the protocol must likewise ensure that, even though the agents pick the plan, they should not be aware about such plan, until they have finished the computation task and returned the result (if employed). We call this property *hiding*. Next, in order to detect the principal's cheating (if any), the agents (after finishing the computation) need to be informed of the plan that was picked (before the computation) for the principal. This must be done in a secure way, so that any attempt to inform the agents of a different plan would be easily caught. We call this the *revealing* property. Finally, if the expected punishment is not credible enough, then the threat of being caught cheating would not deter the principal from deviating. Therefore, we need the last property, called *no revealing*, that requires the execution to give the principal no better benefit than being a honestly sticking to the protocol specification. We summarise these properties below:

- **Correctness**: Honest execution of the protocol must ensure that the plan is picked according to $\Delta(\Omega)$.
- **Hiding**: Before the contract is executed, the agents must know nothing about the plan they have picked for the principal.
- **Revealing**: After the contract is executed, there must be a secure way for the previously picked plan to be revealed to the agents.
- **No cheating**: Suppose that the agents execute the protocol honestly, then the principal receives no better benefit than being a honest principal.

Informally, detection of deviation is a process in which an agent contrasts what it obverses as the principal's behaviour against a plan $\omega \in \Omega$ that was picked by the agents. An inconsistency between the plan and the observed behaviour would indicate to that agent of the principal's deviation. Next, we define $\mathcal{B}_i$ to be the set of possible principal's behaviours *observable* to agent i, along with *consistency relations* $\sim_i$ such that $\omega \sim_i b \Leftrightarrow b \sim_i \omega$ for all pairs $(\omega, b) \in \Omega \times \mathcal{B}_i$, for $i = 1, 2$. Here an observable behaviour is a collection of actions that could be easily and costlessly reproduced (in front of the judges) for verification. For example, the action of assigning a job to an agent, or giving a reward (shown in the bank statement) are observable. In order to capture the principal's benefit from deviation, we need to define its utility for each outcome of the contract execution. Essentially this utility is a function $u_P : \Omega^2 \times (\mathcal{B}_1 \times \mathcal{B}_2 \setminus \{(\emptyset, \emptyset)\}) \to \mathbb{R}$, in which a tuple $(\omega_1, \omega_2, b_1, b_2) \in \Omega^2 \times \mathcal{B}_1 \times \mathcal{B}_2$ indicates the agents' views on the chosen plan (i.e., ω_1 and ω_2), and the principal's actual behaviour from each agent's view (i.e., b_1 and b_2). Here $\emptyset$ denotes the fact that an agent is not employed, and hence $(\emptyset, \emptyset)$ should be excluded, since in that case the principal will need to form a new contract (possibly with different agents).

Our protocol relies heavily on commitments, and hence can be described by a tuple $(\mathcal{S}, \mathcal{C}, \mathcal{D})$ of setup, commit, and decommit algorithms. The protocol starts by a honest generation of a common reference string CK by a reputable trusted third party. This string CK is given securely to all protocol participants. The agents then randomly choose their plan-picking seeds, and produce commitment and opening values for them. The agents then exchange their commitment values via the principal, who is able to intercept and modify such values arbitrarily. After this exchange, the agents proceed to exchange their opening values, again via the principal. The principal, whilst being able to open the seeds committed by the agents, however will not pass on the opening values to the agents (to prevent them from learning each other's seed) until after it has assigned the computation tasks and received results from them. Upon receiving the exchanged opening values, each agent learns the other's seed, then constructs the principal's supposed action plan, and check whether the principal's behaviour agrees with such plan. In our adversary model, we assume that only one party can be malicious/curious at a time. We formalise the security requirements below:

Definition 4.2. *A secure contract implementation mechanism for $\Delta(\Omega)$, $\mathcal{B}_1$, $\mathcal{B}_2$, $\sim_1$, $\sim_2$, and $u_P : \Omega^2 \times (\mathcal{B}_1 \times \mathcal{B}_2 \setminus \{(\emptyset, \emptyset)\}) \to \mathbb{R}$ is a tuple of PPT algorithms $(\mathcal{S}, \mathcal{C}, \mathcal{D})$ satisfying for some negligible function ϵ the following properties:*

- ***Correctness:*** *an honest execution of the protocol must simulate the principal's action plan and the agents' view in an ideal contract execution, i.e., given the*

following experiment for $k \in \mathbb{N}$:

$$\begin{aligned}
&\mathsf{Prot}(\Delta(\Omega), k) : \\
&\quad \mathsf{CK} \leftarrow \mathcal{S}(k) \\
&\quad (c_1, d_1) \leftarrow \mathcal{C}_{\mathsf{CK}}(\Delta(\Omega), k) \\
&\quad (c_2, d_2) \leftarrow \mathcal{C}_{\mathsf{CK}}(\Delta(\Omega), k) \\
&\quad \omega \leftarrow \mathcal{D}_{\mathsf{CK}}(c_1, d_1, c_2, d_2) \\
&\quad \textbf{return } \omega
\end{aligned}$$

it must hold that $\mathsf{Prot}(\Delta(\Omega), k)$ and $\Delta(\Omega)$ are statistically indistinguishable, i.e.,

$$\sup_{\omega \in \Omega} |\Pr[\omega \leftarrow \mathsf{Prot}(\Delta(\Omega), k)] - \Pr[\omega \leftarrow \Delta(\Omega)]| \leq \epsilon(k),$$

- ***Hiding:*** *an agent who receives a commitment value does negligibly better in guessing the eventual action plan ω than the one without such value, i.e., for every PPT algorithm $A = (A_1, A_2)$ (curious agent), there exists a PPT algorithm $A^* = (A_1^*, A_2^*)$ such that*

$$\begin{aligned}
&\Pr\left[\begin{array}{c} \mathsf{CK} \leftarrow \mathcal{S}(k); (c, d) \leftarrow \mathcal{C}_{\mathsf{CK}}(\Delta(\Omega), k); (c', d', m) \leftarrow A_1(\Delta(\Omega), \mathsf{CK}); \\ \omega \leftarrow A_2(c, m, \Delta(\Omega), \mathsf{CK}); \omega' \leftarrow \mathcal{D}_{\mathsf{CK}}(c, d, c', d') : \bot \neq \omega = \omega' \end{array}\right] \\
&- \Pr\left[\begin{array}{c} \mathsf{CK} \leftarrow \mathcal{S}(k); (c, d) \leftarrow \mathcal{C}_{\mathsf{CK}}(\Delta(\Omega), k); (c', d', m) \leftarrow A_1^*(\Delta(\Omega), \mathsf{CK}); \\ \omega^* \leftarrow A_2^*(m, \Delta(\Omega), \mathsf{CK}); \omega' \leftarrow \mathcal{D}_{\mathsf{CK}}(c, d, c', d') : \bot \neq \omega^* = \omega' \end{array}\right] \leq \epsilon(k),
\end{aligned} \tag{4.32}$$

- ***Revealing:*** *after sending a commitment, the principal is only able to convince the other (honest) agent of exactly one action plan ω, i.e., for all PPT algorithm P (cheating principal)*

$$\Pr\left[\begin{array}{c} \mathsf{CK} \leftarrow \mathcal{S}(k); (c, d) \leftarrow \mathcal{C}_{\mathsf{CK}}(\Delta(\Omega), k); (c', d_1, d_2) \leftarrow \\ P(c, d, \Delta(\Omega), \mathsf{CK}); \omega_1 \leftarrow \mathcal{D}_{\mathsf{CK}}(c, d, c', d_1); \\ \omega_2 \leftarrow \mathcal{D}_{\mathsf{CK}}(c, d, c', d_2) : \omega_1 \neq \omega_2 \wedge \omega_1, \omega_2 \neq \bot \end{array}\right] \leq \epsilon(k) \tag{4.33}$$

- ***No cheating:*** *any principal would gain at most negligibly better than a honest*

principal, i.e., given the following experiment

$$
\begin{aligned}
&\Pi^{P}_{\mathsf{Contract\text{-}cheat}}(\Delta(\Omega), k): \\
&\quad \mathsf{CK} \leftarrow \mathcal{S}(k) \\
&\quad (c_1, d_1) \leftarrow \mathcal{C}_{\mathsf{CK}}(\Delta(\Omega), k) \\
&\quad (c_2, d_2) \leftarrow \mathcal{C}_{\mathsf{CK}}(\Delta(\Omega), k) \\
&\quad c_2' \leftarrow P(c_1, c_2) \\
&\quad c_1' \leftarrow P(d_1) \\
&\quad \textbf{if } c_1' = c_1 \vee c_2' = c_2 \textbf{ then return } (\perp, \perp, \emptyset, \emptyset)^{18} \\
&\quad (d_2', b_1, b_2) \leftarrow P(d_2) \\
&\quad \omega_1 \leftarrow \mathcal{D}_{\mathsf{CK}}(c_1, d_1, c_2', d_2') \\
&\quad \omega_2 \leftarrow \mathcal{D}_{\mathsf{CK}}(c_1', d_1', c_2, d_2) \\
&\quad \textbf{return } (\omega_1, \omega_2, b_1, b_2)
\end{aligned}
$$

it must hold that for all stateful PPT algorithms P (cheating principal)

$$
\begin{aligned}
&\mathbb{E}[u_P(\omega_1, \omega_2, b_1, b_2) | (\omega_1, \omega_2, b_1, b_2) \leftarrow \Pi^{P}_{\mathsf{Contract\text{-}cheat}}(\Delta(\Omega), k); (b_1, b_2) \neq (\emptyset, \emptyset)] \\
&- \mathbb{E}[u_P(\omega, \omega, b_1, b_2) | (\omega, b_1, b_2) \leftarrow \mathsf{Prot}(\Delta(\Omega), k); (b_1, b_2) \neq (\emptyset, \emptyset)] \leq \epsilon(k) \quad (4.34)
\end{aligned}
$$

Since the security definition above requires the notion of Ω, $\mathcal{B}_1$, $\mathcal{B}_2$, $\sim_1$, $\sim_2$, and u_P, we also need to specify them before designing our protocol. Note that since we aim at designing a protocol that assumes no communication among the agents, our protocol only needs to serve contracts against non-colluding agents. Therefore we do not include action plans involving auditing when returned results are different (β-scheme) or identical (ν-scheme), but instead focus on optimal contracts for non-colluding agents as in Propositions 4.1, 4.3, and 4.5. In other words, we represent the set of plans as

$$
\Omega = \{\{A_1, audit\}, \{A_1, reward\}, \{A_2, audit\}, \{A_2, reward\}, \{A_1, A_2, reward\}, \perp\}. \quad (4.35)
$$

Particularly, a plan must specify the employed agent(s), as well as what to do when the results are returned. The $\{A_1, A_2, reward\}$ plan implies two-agent employment, and that if the returned results are the same, then both agents are rewarded, or punished

[18] This is to prevent the cheating principal P from returning a commitment back to the same agent, pretending that it comes from another one, as that would clearly nullify all the seeds for selecting $\omega_i \in \Omega$.

otherwise. Similarly we have the set of agents' view on the principal's behaviours:

$$\mathcal{B}_1 = \{\{A_1, audit\}, \{A_1, reward\}, \emptyset\} \quad \text{and} \quad \mathcal{B}_2 = \{\{A_2, audit\}, \{A_2, reward\}, \emptyset\} \tag{4.36}$$

Our construction of $\mathcal{B}_i$ relies on the *assumption* that the acts of auditing and blind rewarding are distinguishable and verifiable to the agents. Note that we do not consider *punish* as an observable action because we only aim to protect honest agents from being cheated by the principal. Next, the consistency relations follow straightforwardly, i.e., for all $b_i \in \mathcal{B}_i$ and $\omega \in \Omega$:

$$b_i \sim_i \omega \iff \omega \sim_i b_i \iff \begin{cases} b_i = \emptyset \wedge A_i \notin \omega, \text{ or} \\ b_i \neq \emptyset \wedge b_i \subset \omega \end{cases} \tag{4.37}$$

Interestingly, it is easy to notice that for every $\omega \in \Omega$ and $i \in \{1, 2\}$, there is exactly one $b \in \mathcal{B}_i$ such that $b \sim_i \omega$. Nevertheless, the principal's utility is then either its cost of executing the contract, or a fine from the court if being caught cheating by at least one of the agents, i.e.,

$$u_P(\omega_1, \omega_2, b_1, b_2) = \begin{cases} -r - \max(r, \gamma) \text{ if } b_1 \not\sim_1 \omega_1 \vee b_2 \not\sim_2 \omega_2, \text{ or} \\ \text{contract executing cost otherwise.} \end{cases} \tag{4.38}$$

We are now ready to construct our contract implementation protocol. For each probability distribution $\Delta(\Omega)$ assume that there exists a PPT *contract-generation* algorithm $\mathsf{Gen}_{\Delta(\Omega)}(\cdot)$ which efficiently samples $\Delta(\Omega)$, that is, there exists a negligible function ϵ_{G} such that

$$\sup_{o \in \Omega} \left| \Pr\left[r \leftarrow_{\$} \{0,1\}^k; \omega \leftarrow \mathsf{Gen}_{\Delta(\Omega)}(r)\right] - \Pr\left[\omega \leftarrow \Delta(\Omega)\right] \right| \leq \epsilon_{\mathsf{G}}(k). \tag{4.39}$$

Whilst the protocol construction can be seen in Figure 4.5, its security is described in Proposition 4.9 and its proof.

Proposition 4.9. *Let* $(\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open})$ *be a non-malleable commitment scheme. Let* Ω*,* $\mathcal{B}_i$*,* $\sim_i$ *and* u_P *be as defined in* (4.35)*,* (4.36)*,* (4.37)*, and* (4.38)*, respectively. Let* $\Delta(\Omega)$ *be a probability distribution over* Ω*, and assume existence of* $\mathsf{Gen}_{\Delta(\Omega)}$ *as in* (4.39)*. Then there exists a secure contract implementation mechanism for* $\Delta(\Omega)$*,* $\mathcal{B}_i$*,* $\sim_i$ *and* u_P*.*

Proof. We show that $\mathsf{ContractProtocol}$ satisfies all the properties of a secure contract implementation mechanism. The proof of correctness is rather trivial as it comes

Protocol ContractProtocol

proc $\mathcal{S}(k)$:
  $\mathsf{CK} \leftarrow \mathsf{Setup}(k)$
  return(CK)

proc $\mathcal{C}_{\mathsf{CK}}(\Delta(\Omega), k)$:
  $r \leftarrow_{\$} \{0,1\}^k$
  $(c, d) \leftarrow \mathsf{Commit}_{\mathsf{CK}}(r)$
  return(c, d)

proc $\mathcal{D}_{\mathsf{CK}}(c_1, d_1, c_2, d_2)$:
  $r_1 \leftarrow \mathsf{Open}_{\mathsf{CK}}(c_1, d_1)$
  $r_2 \leftarrow \mathsf{Open}_{\mathsf{CK}}(c_2, d_2)$
  if $r_1 = \perp \vee r_2 = \perp$ **then**
    return $\perp$
  $r \leftarrow r_1 \oplus r_2$
  $\omega \leftarrow \mathsf{Gen}_{\Delta(\Omega)}(r)$
  return ω

Figure 4.5: Communication protocol for the contract

straightforwardly from the design of the contract generation algorithm $\mathsf{Gen}_{\Delta(\Omega)}$. Indeed, the correctness experiment $\mathsf{Prot}(\mathsf{CK}, k)$ produces two uniformly random nonces $r_1, r_2 \in \{0,1\}^k$, and hence $\{0,1\}^k \ni r = r_1 \oplus r_2$ is also uniformly random. Since $\mathcal{D}_{\mathsf{CK}}(c_1, d_1, c_2, d_2)$ uses $\mathsf{Gen}_{\Delta(\Omega)}$ to generate ω, (4.39) directly implies the correctness property.

For the hiding property, we assume that there exists a PPT algorithm A such that no PPT algorithm A^* can preserve (4.32) and construct an adversary A' against the hiding property of the commitment scheme. First, given the existence of $A = (A_1, A_2)$, let $A^* = (A_1^*, A_2^*)$ such that $A_1^* = A_1$ and $A_2^*(m, \Delta(\Omega), \mathsf{CK}) = A_2(c^*, m, \Delta(\Omega), \mathsf{CK})$, where $(c^*, d^*) \leftarrow \mathsf{Commit}_{\mathsf{CK}}(0^k)$. The fact that (4.32) does not hold thus implies

$$\Pr\left[\begin{array}{c}\mathsf{CK} \leftarrow \mathcal{S}(k); (c,d) \leftarrow \mathcal{C}_{\mathsf{CK}}(\Delta(\Omega), k); (c', d', m) \leftarrow A_1(\Delta(\Omega), \mathsf{CK}); \\ \omega \leftarrow A_2(c, m, \Delta(\Omega), \mathsf{CK}); \omega' \leftarrow \mathcal{D}_{\mathsf{CK}}(c, d, c', d') : \perp \neq \omega = \omega'\end{array}\right]$$
$$-\Pr\left[\begin{array}{c}\mathsf{CK} \leftarrow \mathcal{S}(k); (c,d) \leftarrow \mathcal{C}_{\mathsf{CK}}(\Delta(\Omega), k); (c', d', m) \leftarrow A_1(\Delta(\Omega), \mathsf{CK}); \\ \omega^* \leftarrow A_2(c^*, m, \Delta(\Omega), \mathsf{CK}); \omega' \leftarrow \mathcal{D}_{\mathsf{CK}}(c, d, c', d') : \perp \neq \omega^* = \omega'\end{array}\right] = \epsilon'(k) \quad (4.40)$$

for some non-negligible function ϵ'. As a result, the attacker A' against the commitment scheme proceeds as in the following experiment:

$$\Pi^{A'}_{\mathsf{NM\text{-}hide}} :$$

$$
\begin{aligned}
&\mathsf{CK} \leftarrow \mathsf{Setup}(k)\\
&(c, d) \leftarrow \mathcal{C}_{\mathsf{CK}}(\Delta(\Omega))\\
&m_0 \leftarrow \mathsf{Open}_{\mathsf{CK}}(c, d)\\
&m_1 \leftarrow 0^k\\
&(c', d', m) \leftarrow A_1(\Delta(\Omega), \mathsf{CK})\\
&\omega \leftarrow \mathcal{D}(c, d, c', d')\\
&b \leftarrow_{\$} \{0, 1\}\\
&(c^*, d^*) \leftarrow \mathsf{Commit}_{\mathsf{CK}}(m_b)\\
&\omega' \leftarrow A_2(c^*, m, \Delta(\Omega), \mathsf{CK})\\
&\textbf{if } \omega = \omega' \textbf{ then } b' \leftarrow 0\\
&\quad \textbf{else } b \leftarrow 1\\
&\textbf{return } b = b'
\end{aligned}
$$

Denote (4.40) in short as $p - q = \epsilon'(k)$ we notice that when $b = 0$, A_2's view in $\Pi^{A'}_{\mathsf{NM\text{-}hide}}$ is identical to the experiment associated with p. Thus, in that case $\Pr[\omega = \omega'] = p$. Similarly, when $b = 1$ we have $\Pr[\omega \neq \omega'] = 1 - q$. Therefore

$$
\begin{aligned}
\Pr[\textbf{true} \leftarrow \Pi^{A'}_{\mathsf{NM\text{-}hide}}] &= \Pr[b = 0, b' = 0] + \Pr[b = 1, b' = 1]\\
&= \Pr[b = 0]\Pr[b' = 0 | b = 0] + \Pr[b = 1]\Pr[b' = 1 | b = 1]\\
&= \frac{1}{2}p + \frac{1}{2}(1 - q)\\
&= \frac{1}{2}(q + \epsilon'(k)) + \frac{1}{2}(1 - q) = \frac{1}{2} + \frac{1}{2}\epsilon'(k)
\end{aligned}
$$

which contradicts with the hiding property of commitment since $\epsilon'(k)/2$ is non-negligible. This thus proves that the hiding property of the commitment scheme is broken, which contradicts with our assumption on its security.

In a similar way, we show that an attacker P on the revealing property of contract implementation can be used to construct an attacker P' on the *binding* property of the commitment scheme. The proof, however, is rather straightforward:

$$
\begin{aligned}
&\Pi^{P'}_{\mathsf{NM\text{-}bind}}:\\
&\mathsf{CK} \leftarrow \mathsf{Setup}(k)\\
&(c, d) \leftarrow \mathcal{C}_{\mathsf{CK}}(\Delta(\Omega))\\
&(c', d_1, d_2) \leftarrow P(c, \Delta(\Omega), \mathsf{CK})\\
&m_1 \leftarrow \mathsf{Open}_{\mathsf{CK}}(c', d_1)
\end{aligned}
$$

$$m_2 \leftarrow \mathsf{Open}_{\mathsf{CK}}(c', d_2)$$
$$\textbf{return } m_1 \neq m_2 \wedge m_1, m_2 \neq \perp$$

By looking at the construction of $\mathcal{D}$ we notice that, $\omega_1, \omega_2 \neq \perp$ from the experiment in (4.33) implies $m_1, m_2 \neq \perp$. Similarly, the correctness of the commitment scheme guarantees that $\omega_1 \neq \omega_2$ implies $m_1 \neq m_2$. Thus, if P succeeds in breaking the revealing property with non-negligible chance, then $\Pi^{P'}_{\mathsf{NM\text{-}bind}}$ returns true with the same probability, i.e., the binding property of the commitment scheme is broken.

What remains is to prove that the principal only gains negligibly better by cheating. To show this, we notice that following (4.38), the principal's utility is essentially its cost in executing the contract, which includes the reward to the agents and the cost of auditing. Denote by $u_1(\omega_1, b_1)$ and $u_2(\omega_2, b_2)$ the principal's cost with respect to agent 1 and 2, respectively. Since the contracts in Ω does not involve auditing both agents, meaning that the auditing cost is always spent with respect to exactly one agent, therefore we have:

$$u_P(\omega_1, \omega_2, b_1, b_2) = u_1(\omega_1, b_1) + u_2(\omega_2, b_2)$$

Consider the experiment $\Pi^{P}_{\mathsf{Contract\text{-}cheat}}(\Delta(\Omega), k)$ and utility function (4.38), we notice that the principal would avoid making either $\omega_1 = \perp$ or $\omega_2 = \perp$, as it would certainly receive utility $-r - \max(r, \gamma)$ which is the maximum realisable expense for executing the contract. It would also avoid making either $b_1 \not\sim_1 \omega_1$ or $b_2 \not\sim_2 \omega_2$ for the same reason. Let $r_1 \leftarrow \mathsf{Open}_{\mathsf{CK}}(c_1, d_1)$, $r'_2 \leftarrow \mathsf{Open}_{\mathsf{CK}}(c'_2, d'_2)$. Due to the non-malleability property of commitment, r_1 and r'_2 are statistically independent, and since r_1 is generated uniformly randomly by $\mathcal{C}_{\mathsf{CK}}(\Delta(\Omega), k)$, therefore $r_1 \oplus r'_2$ is statistically indistinguishable from uniform randomness. Consequently this implies that $\omega_1 \leftarrow \mathcal{D}_{\mathsf{CK}}(c_1, d_1, c'_2, d'_2)$ is statistically indistinguishable from $\Delta(\Omega)$. A similar argument also applies to $\omega_2 \leftarrow \mathcal{D}_{\mathsf{CK}}(c'_1, d'_1, c_2, d_2)$. We thus have for some negligible function ϵ_2 such that:

$$\begin{aligned}
&\mathbb{E}\left[u_P(\omega_1, \omega_2, b_1, b_2) | (\omega_1, \omega_2, b_1, b_2) \leftarrow \Pi^{P}_{\mathsf{Contract\text{-}cheat}}(\Delta(\Omega), k); (b_1, b_2) \neq (\emptyset, \emptyset)\right] \\
&= \sum_{(\omega'_1, \omega'_2, b'_1, b'_2) \neq (\cdot, \cdot, \emptyset, \emptyset)} u_P(\omega'_1, \omega'_2, b'_1, b'_2) \frac{\Pr[(\omega_1, \omega_2, b_1, b_2) = (\omega'_1, \omega'_2, b'_1, b'_2)]}{\Pr[(b_1, b_2) \neq (\emptyset, \emptyset)]} \\
&\leq \sum_{(\omega'_1, \omega'_2, b'_1, b'_2) \neq (\cdot, \cdot, \emptyset, \emptyset)} u_P(\omega'_1, \omega'_2, b'_1, b'_2) \Pr[(\omega_1, \omega_2, b_1, b_2) = (\omega'_1, \omega'_2, b'_1, b'_2)] \\
&= \sum_{(\omega'_1, \omega'_2, b'_1, b'_2) \neq (\cdot, \cdot, \emptyset, \emptyset)} (u_1(\omega'_1, b'_1) + u_2(\omega'_2, b'_2)) \Pr[(\omega_1, \omega_2, b_1, b_2) = (\omega'_1, \omega'_2, b'_1, b'_2)] \quad (4.41)
\end{aligned}$$

$$= \sum_{\omega_1',\omega_2',b_1',b_2'} (u_1(\omega_1', b_1') + u_2(\omega_2', b_2')) \Pr\left[(\omega_1, \omega_2, b_1, b_2) = (\omega_1', \omega_2', b_1', b_2')\right] \tag{4.42}$$

$$= \sum_{\omega_1',b_1'} u_1(\omega_1', b_1') \Pr\left[(\omega_1, b_1) = (\omega_1', b_1')\right] + \sum_{\omega_2',b_2'} u_1(\omega_2', b_2') \Pr\left[(\omega_2, b_2) = (\omega_2', b_2')\right] \tag{4.43}$$

$$= \sum_{\omega_1' \sim_1 b_1'} u_1(\omega_1', b_1') \Pr\left[(\omega_1, b_1) = (\omega_1', b_1')\right] + \sum_{\omega_2' \sim_2 b_2'} u_1(\omega_2', b_2') \Pr\left[(\omega_2, b_2) = (\omega_2', b_2')\right] \tag{4.44}$$

$$= \mathbb{E}[u_P(\omega, \omega, b_1, b_2) | (\omega, b_1, b_2) \leftarrow \mathsf{Prot}(\Delta(\Omega), k); (b_1, b_2) \neq (\emptyset, \emptyset)] + \epsilon_2(k).$$

For ease of comprehension, we note that (4.42) comes from (4.41) because $u_1(\cdot, \emptyset) = u_2(\cdot, \emptyset) = 0$. Also, (4.43) yields (4.44) because earlier we argue that the principal would let neither $b_1 \not\sim_1 \omega_1$ nor $b_2 \not\sim_2 \omega_2$. This completes the proof of the security of ContractProtocol. □

4.9 Conclusion

In this chapter, we provide an incentive analysis of outsourced computation with non-malicious but selfish utility-maximising agents. We design contracts that minimise the expected cost of the outsourcer whilst ensuring participation and honesty of computing agents. We incorporate important real-world restrictions, in that the outsourcer can only levy a restricted fine on dishonest agents and that auditing can be costly and/or limited. We allow partial outsourcing, direct auditing and auditing through redundancy, i.e., employing multiple agents and comparing the results, and optimised the utility of the outsourcer among all hybrid possibilities.

We observe that outsourcing all or none of the tasks is optimal (and not partial outsourcing). We show that when the enforceable fine is restricted, achieving honest computation may still be feasible by appropriately increasing the reward above the sheer cost of honest computation. We demonstrate that when auditing is more expensive than the cost of honest computation, redundancy scheme is always the preferred method, and when the auditing cost is less than half of the cost of honest computation, independent auditing is preferable. When the cost of auditing is between half and the full cost of honest computation, the preferred method depends on the maximum enforceable fine: for large enforceable fines, redundancy scheme is preferred despite the fact that it is more expensive "per use" than independent auditing, since owing to its higher effectiveness, it can be used more sparingly. We established the global optimality of contracts involving at most two agents among any arbitrary number of agents as far

as implementing honesty as a Nash Equilibrium is aimed for.

Another focus of this work is on the effect of side information and collusion on the optimal contracts involving a hybrid of direct computation and redundancy scheme (duplication of the same task to two agents and comparing the returned results). In particular, we explicitly developed conditions in which the redundancy scheme fails to be the preferred method and conditions in which it will be the preferred method even under such adverse conditions. Notably, we showed that that making collusion a less attractive equilibrium is *not* an effective way at all to save the redundancy scheme in the face of collusion. Instead, an effective way is bounty-like schemes that attempt to make collusion a dis-equilibrium. Overall, we noted that preference for redundancy in the presence of side information or collusion occurs for high values of auditing cost (expected), and low values of maximum enforceable fines, where the latter is in sharp contrast with the cases that collusion or side information is absent. This work in part provided insights on potentials and limitations of redundancy scheme as a method of auditing. Finally, we present a light-weight cryptographic implementation of our contracts that provides mutual affirmation on proper execution of the agreed terms and conditions.

Future directions. Our work opens a number of potential avenues for future investigation. One of the major scenarios which we have simplified in this work is the possible interactions among the agents. Here we assume that agents share accurate information about their state with respect to the job assignments to each others, and then each individually and independently decides its action. However, the agents may be able to deceive their peers by giving them wrong signals about their state with the objective of winning the bounty. Also, we assume cheating agents, although able to collude, cannot have a means of commitment among themselves. If enforceable commitments among colluding agents are assumed, the analysis can become more complicated: the agents may agree to pass the honest result to one another, or intentionally plan for one of them to get the bounty, only to share it among themselves later. In terms of global optimality, we established that when agents are non-colluding and non-communicating, the optimal contracts developed assuming at most two agents per each task are in fact globally optimal among all contracts involving any number of agents per task. However, in the presence of information leakage and collusion, this becomes more challenging, as more parameters (e.g., the bounty) can be involved to build contracts. This enables further investigation which are interesting as future research.

Chapter 5

Rational Security for Unauthenticated Communication

In the context of secure communication, cryptographic techniques often operate on the basis of entity authentication. On the other hand, studies of security for unauthenticated communication have often not been well motivated, and have thus received little attention from the research community. This may be partly because cryptographic definitions of security make it difficult to find an acceptable solution, but also due to an argument that security, as a service, is not for free, whereas authentication carries a cost for its own infrastructure.

In this chapter we provide an initial study of the security of unauthenticated communications under the rational security model. We investigate a simple but prevalent type of communication, namely, the *query-response* protocol, in which one party sends a request for information, and the other replies. With a meaningful relaxation of attack called *online man-in-the-middle* on query-response communication, we show a positive theoretical result: even without any mean of authentication, such attacks can always be rationally discouraged. Our notion of security is represented under the form of a 3-player game perfect Bayesian equilibrium, in which two communicating peers follow a prescribed protocol, whilst the adversary choose to not perform the designated attack.

We justify in the chapter our assumptions and definitions in relation to real life scenarios, as well as discuss matters on the practicality of our theoretical result. We hope that the work presented in this chapter motivates further research on rational security of communication protocols, in particular more elaborate unauthenticated communication protocols.

5.1 Introduction

Due to physical and geographical separation, secure communication is vital in our inter-networked world. Various aspects of security have been established for this need, e.g., confidentiality, integrity, non-repudiation. Security services, such as encryption and message authentication, that can offer these aspects often operate on top of a mechanism that guarantees authenticity/correct identification of communicating parties. Typically, a secure communication (e.g. TLS-protected) between two networked parties consists of three phase [44]. In the beginning, the parties present their identities to one another. Next, they execute a key exchange protocol to establish a common session key. Lastly, security services such as encryption and message authentication are employed, using the common key, to protect messages in transmission. While the common key (along with encryption and message authentication) guarantees that a message on the line cannot be read or tampered with by anyone other than the sender and intended receiver, it cannot assure who these sender and receiver actually are. Therefore, an attacker can easily pretend to be the sender or receiver. Such assurance can only be accomplished during the first phase, i.e., authentication. As a result, entity authentication mechanisms become the "Achilles' heel" of security systems, and in their absence security services can seldom be considered fully effective.

Authentication in general is not an easy task. The most notable authentication systems are Public Key Infrastructures (PKIs), but they mostly target communications that create monetary or sensitive effects, such as banking, e-commerce or military. Another example is Web of Trust (WoT), which can offer authentication for more neutral communication, but is less secure and scalable, as it depends on the efforts of selfish individuals within the WoT. When it is assumed that attacks during the first connection between two parties are unlikely, then the so-called *leap-of-faith* (LoF) [112] can be used to offer "quasi-authentication". The parties will make a leap of faith that no attack occurs, and remember each other's authenticating credential, so that in the future mutual authentication can be achieved.

Undoubtedly the vast majority of Internet communication is unauthenticated, e.g., WWW surfing, peer-to-peer networks such as BitTorrent. On the one hand, it is arguable that these communications are often not worth protecting due to their low values and sensitivity. On the other hand, the number of entities involved are beyond the managerial capacity of nowadays infrastructures. We are thus intrigued by the question: "What security can be achieved in the presence of an attacker, without authentication?", as similarly asked by Barak et al. [10].

The above question has unfortunately received little attention from the research

community. One obvious reason is that, since definitions of security adopted in conventional cryptographic and network security research are so strong, in the absence of authentication they are simply too difficult, or even impossible to achieve. Notable examples are the concepts of matching conversations and mutual authentication by Bellare and Rogaway [18]. Moreover, popular adversarial models, e.g., PPT adversaries in provable security, are often too powerful and much more highly motivated than what one would encounter in most real-life cases.

Without authentication, two-party communication may suffer from two important attacks: *impersonation* and *man-in-the-middle* (MitM). In impersonation attacks the adversary pretends to be one of the parties, and communicates with the other. MitM attacks, on the other hand, allow the adversary to control the communication between the two parties. Here the MitM attacker has an advantage in communicating with each party, as it may receive useful information from the other. It is thus natural to assume that the latter are more powerful and hence more harmful than the former.

It is a theoretical implication that the lack of authentication directly leads to the threat of impersonation, which is unavoidable. We therefore focus, instead, on what can be done against MitM attacks. Specifically, we investigate a proliferated form of communication: request-reply. Our works involve two steps. First we specialise the generic notion of MitM attacks from cryptographic/formal method literatures, so as emphasise the clear distinction between these and impersonation attacks in the context of request-reply communication. Then, we devise an economic model involving communicating parties and a potential adversary, within which we show that there is always a mechanism that discourages MitM attacks. Of course the adversary may still perform impersonation, but in our argument, that is the best it could do.

Our work is organised as follows. In Section 5.2 we review existing related literature, which indicates a lack of solutions for unauthenticated communication. Then we establish our framework with a redefinition of MitM attacks in Section 5.3, and an economic model in Section 5.4 representing the security problem. Under this model, we progress by showing in Section 5.5 that there exists an equilibrium in which the adversary does not attack. Section 5.6 shows a cryptographic implementation of such equilibrium, thus making it a feasible solution. Although our solution is a proof-of-concept that unauthenticated security is nontrivial, we nevertheless discuss concerns regarding its practicality in Section 5.7.

5.2 Related Work

Man-in-the-middle attacks have always been a central problem in communication security, both practically and theoretically. Formal treatments of such attacks have been considered in several different ways. Most notable ones are summarised as below.

Ping-pong protocols. Dolev et al. [46, 47] defines successful attackers as ones who can revert a function evaluation on a message sent over the communication channel. Their focus on *ping-pong* protocols is particularly related to our context, as any protocol implementing a query-response communication is of type *ping-pong*, i.e., the communication is stateless. However, their definition has the flavour of formal-method security, and thus it alleviates many important low-level cryptographic requirements which may be nontrivial to satisfy. Also, it mostly serves analysis of attacks on confidentiality, and hence assumes authentication in the first place. Instead, we look at a more natural problem (cryptographically) associated with man-in-the-middle attacks: lack of (even weak) authentication.

Oracled-based security. Probably the most popular model of man-in-the-middle attacks is via oracle-based models. Particularly, to model a security problem, one would set up a context, with necessary information (keys, nonces, etc.) ready for the communication, except that there are no communicating parties. Instead, all these information are embedded into one or more input/output *oracle(s)*, with which an adversary can interact with. The adversary is successful if, given interaction with the oracle(s), it is able to produce some legitimate messages. Man-in-the-middle attacks are thus effectively captured, in that one can think of the oracle(s) as one endpoint where the adversary receives helps in order to communicate with the other.

Oracled-based models are powerful in that the adversary is not bound to any communication protocol, and is thus free on what it can do (as there is no worry about attack detection). Meanwhile, each model is normally used to capture a very specific and narrow security requirement, as otherwise the formal definition might become too complicated to express. In fact, different security objectives have been captured using oracle-based models, such as encryption (public-key [119], private-key [79]), message authentication [16], digital signature [61].

Non-malleability. The more intuitive concept, but also more cumbersome to work with, is non-malleability. Informally, a communication protocol is non-malleable if, each party's interaction with the adversary A can be simulated by an isolated algorithm that does not interact with anyone. In other words, any information the adver-

sary receives from the communication is not useful, as whatever it can do with such information, it can also do without. This thus violates the common understanding of man-in-the-middle attacks, in which an adversary must rely on manipulating of communication messages. Non-malleability is introduced by Dolev et al. [45], applied to several objectives, including commitment, encryption, and zero-knowledge proofs.

Unauthenticated communication. There have been some attempts to study security for unauthenticated communication. In [10] Barak et al. consider the problem of multiparty computation, which usually require pairwise secure communication among parties involved. They study the problem when there is completely no authentication. Their idea is first to let parties exchange their public keys, then executing a secure multiparty protocol (SMPC). Relying on the power of SMPC, this mechanism ensures that the adversary can only perform sequential impersonations, and not meaningful man-in-the-middle attacks. For example, in attacking the communication between Alice and Bob, the protocol ensures that, if the adversary wants to use information obtained from Alice in talking to Bob, it must first completely finish a protocol execution with Alice. In other words, the adversary cannot maintain concurrent conversations with both sides and pass on messages online. In another work, Maurer and Wolf [96] propose key agreement protocols over unauthenticated public channels, which is secure if the adversary cannot simulate either party and its knowledge. Finally, Pham and Aura [113] devises a framework to analyse whether LoF is an acceptable replacement for authentication without compromising too much security. Note that these solutions still assume some form of correlation in protocol participant's knowledge as well as restriction in adversary's capabilities.

Our contributions. Our work coincides with that of Barak et al. [10] in the research question as well as objective, but using a different approach, and hence giving a solution with different properties. Particularly, we consider the question of whether non-trivial yet meaningful security can be achieved in completely unauthenticated settings. To answer this, we propose the notion of online-man-in-the-middle (oMitM) attack, which in turn yields a relaxed notion of security. In fact, our concept of oMitM attack/security is the same as that developed by Barak et al. More specifically, it guarantees that either the adversary will not tamper with the communication, or must do so sequentially (e.g., first finishing all communication with Alice, then starting with Bob, not concurrently).

Our argument is that while Barak et al. provides a cryptographic solution that satisfies the relaxed security, its use of SMPC techniques make it practically unfavourable, as it requires considerable efforts, only works well with static computation, and is not

compatible with modules unfamiliar with SMPC [110]. Instead, we seek a solution that operates totally beneath the actual communication. At this stage, our goal is to provide a solution for query-response protocols, the most primal form of communication in the Internet.

In addition to a proposal of relaxed security, we develop a game-theoretic model to study the conditions under which the adversary does not have an incentive to carry oMitM attacks, and consider this as satisfaction of security. Our notion of security is thus rational, as opposed to cryptographic one by Barak et al. We then develop a cryptographic protocol operating beneath the query-response communications that forces the adversary into playing a game we design for him and the two endpoints. By playing this game in our designated equilibrium points, the adversary is incentivised to at most perform sequential impersonation, not oMitM attacks. Compared to SMPC, our solution is cheaper in complexity, and fully supports all dynamic changes in the application-level communication.

5.3 The Nature of MitM Attacks

In this section we formally define the notion of communication which is convenient for our study, which is then followed by the (weaker) notion of man-in-the-middle attack. We also provide explanation of why we believe that such attack can be effectively and efficiently discouraged in the lack of authentication facilities.

5.3.1 Defining Query-Response MitM Attacks

As the first study in the economics of unauthenticated communication, we focus on the most primal and typical form of communication, i.e., *single-round query-response*. A single-round query-response communication consists of two parties, a client and a server. During the course of communication, the client sends a query $q \in \mathcal{Q}$ to the server who, upon receipt of q, would send back a response $r \in \mathcal{R}$. Because the communication is considered single-rounded, this process happens only once from the client's view. We start by defining the queries and responses which are meaningful to the client and the server:

Definition 5.1. *A* query-response space *is a tuple $\langle \mathcal{Q}, \mathcal{R}, F \rangle$, where $\mathcal{Q} \not\ni \perp$ and $\mathcal{R} \not\ni \perp$ are finite sets of strings with polynomially-bounded lengths , and F is the set of polynomial-time computable functions from $\mathcal{Q}$ to $\mathcal{R}$.*

In the above definition, the finiteness of $\mathcal{Q}$ and $\mathcal{R}$ indicates that the client and the server can recognise if a query/response is valid, e.g., a well-formatted HTTP

query/response. Any other malformed query/response would be treated as $\perp$. In the next step we define *communication transcripts*, which is the central concept in formalisation of attacks. In short, a communication transcript consists of messages exchanged among parties during a protocol execution. We formalise this notion below:

Definition 5.2. *A* messaging event *is a tuple* (A, B, m, s, t) *indicating that party* A *starts sending a message* m *at time* s *and party* B *receives it at time* t*. With respect to a query-response space* $\langle \mathcal{Q}, \mathcal{R}, F \rangle$*, a* communication transcript *is a finite sequence of messaging events* $\langle (A_i, B_i, m_i, s_i, t_i) \rangle_n$ *where* $m_i \in \mathcal{Q} \cup \mathcal{R} \cup \{\perp\}$ *and* $s_i \leq t_i$*.*

In this case the notion of "message" is context-specific, and thus is not necessarily a physical message being sent over the network. For example, a HTTP request message is sent from the client to the server after several IP packets to establish a TCP connection, followed by a few more TCP segments if the request content is large enough. However, in the context of HTTP, all these exchanges are counted as one message being sent. At this moment, we assume that knowledge about a message is always atomic, that is, it is either known in full or not at all. Note also that a messaging event (A, B, m, t) has a universal meaning, rather than from any party's viewpoint. We are now ready to define a protocol for query-response communication:

Definition 5.3. *Let* $\langle \mathcal{Q}, \mathcal{R}, F \rangle$ *be a query-response space. A single-round protocol implementing* $\langle \mathcal{Q}, \mathcal{R}, F \rangle$ *is a tuple* $\langle \mathsf{Qry}, \mathsf{Res} \rangle$ *such that:*

- Qry *is a PPT algorithm that takes as input a query* $q \in \mathcal{Q}$*,*
- Res *is a PPT algorithm that takes as input* $f \in F$*,*
- *let* $C := \mathsf{Qry}(q)$ *and* $S := \mathsf{Res}(f)$*, denote by* $\mathsf{transcript}$ *the communication transcript w.r.t.* $\langle \mathcal{Q}, \mathcal{R}, F \rangle$ *of an execution of* C *and* S*, then* $\forall q \in \mathcal{Q}$ *and* $\forall f \in F$*:*

$$\Pr[\exists\ (C, S, \overline{q}, s_1, t_1), (S, C, r_{\overline{q}}, s_2, t_2) \in \mathsf{transcript} : \overline{q} = q \wedge r_{\overline{q}} = f(q)] = 1 \quad (5.1)$$

where $\overline{q}$ *and* $r_{\overline{q}}$ *denote a query made by* C *and the corresponding response it received, respectively.*

Regarding Definition 5.3, one can think of Qry and Res as the protocol implementation for the client and the server, respectively. An example of Qry, in reality, is a WWW browser, and the corresponding Res is the WWW server. A client will think of a query $q \in \mathcal{Q}$ and enter it on the browser. The browser interacts (in some way) with the server, and it presents a response r to the client. For correctness, the requirement

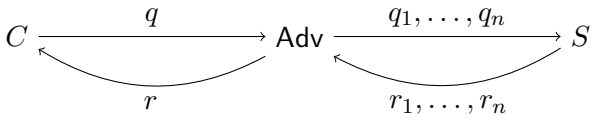


Figure 5.1: Attacks on a query-response conversation

(5.1) makes sure that when there is no attacks or errors, the server S must receive the correct query q, and the client C must receive the correct response $r = f(q)$.

The protocol defined in Definition 5.3 suffers from attacks as presented in Figure 5.1. In this case there is potentially an adversary sitting in the middle of the conversation. The simplest form of attacks is *impersonation*, in which the adversary simply drops the conversation with one end and interacts only with the other. The more sophisticated attacks are *man-in-the-middle* (MitM). In particular, the adversary captures traffic from C representing a query $q \in \mathcal{Q}$, and then makes a number of queries to S. By receiving the traffic containing the replies, the adversary Adv somehow crafts a traffic transmitting a response $r \in \mathcal{R}$ back to C. As benefits, the adversary for example learns the content of q or $f(q)$, or is able to make r different from $f(q)$, and so on.

Without authentication, neither impersonation nor MitM attacks can be discouraged in cryptographic sense. This is because the adversary Adv has exactly the same knowledge as the server S, except that of f, which it can replace by some arbitrary $f' \in F$. Instead, let us consider a relaxing, but also natural form of attacks, which we call *online man-in-the-middle* (oMitM): the attack must necessarily yield an information flow from C to S and then back to C, both via the adversary Adv. In other words, the adversary must rely on communication with each end in talking with the other.

A successful oMitM attack consequently requires both ends to be concurrently active for at least some period of time. By "active" we mean that a party is engaged in a protocol execution and has not finished it yet. The intuition is clear: if concurrent activeness is not required, then any one-way/two-way impersonation would become a valid attack, which is unavoidable and thus uninteresting. Indeed, referring to Figure 5.1, Adv is only considered a successful oMitM attacker if some q_i correlate(s) with q and r correlates with the corresponding r_i. Due to atomicity of message knowledge, this means that at least some query q_i must be made after Adv receives q. Similarly, the response r must be sent after Adv receives $r_i = f(q_i)$ from S. In other words, for two messaging events $(C, \mathsf{Adv}, q, s_q, t_q)$ and $(\mathsf{Adv}, C, r, s_r, t_r)$, there exist messaging events $(\mathsf{Adv}, S, q_i, s_1, t_1)$ and $(S, \mathsf{Adv}, r_i, s_2, t_2)$ such that $t_q < s_1 < t_2 < s_r$. The flow of information is thus from q to q_i, r_i, and finally to r. Since C and S must be active within $[s_q, t_r]$ and $[t_1, s_2]$ respectively, the concurrent activeness period is at least $[t_1, s_2]$. We visualise this in Figure 5.2 and summarise the notion of oMitM adversary below.

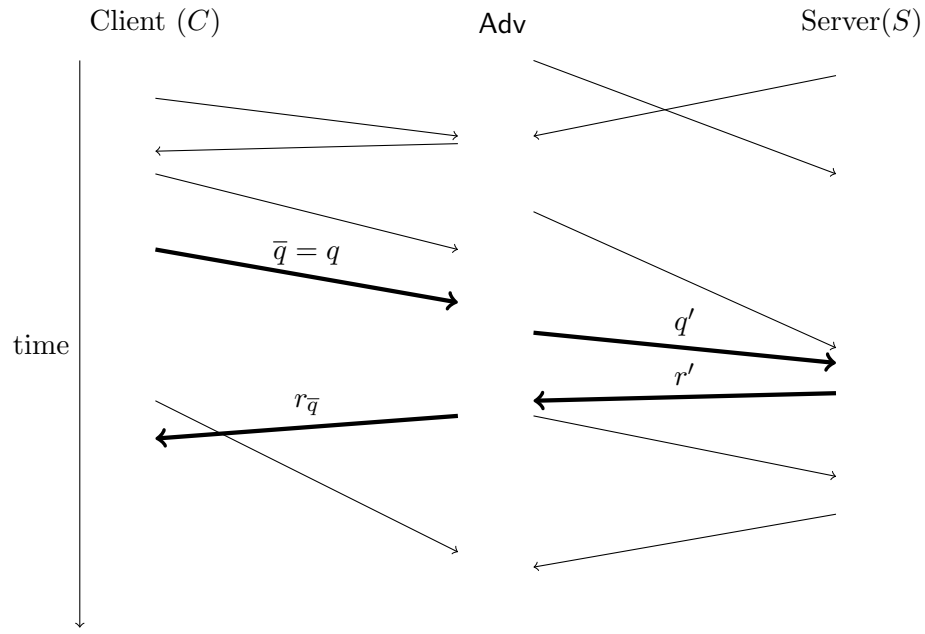


Figure 5.2: Example communication transcript with successful oMitM attack

In overall, our relaxation of security makes sense in that, a non-online MitM attacker, although is able to modify the traffic, cannot exploit the power of a man in the middle: using the convenience in access to information from one side in communicating with the other. In this case, such attacker might equivalently send a random response back to $C := \mathsf{Qry}(q)$ without even talking to $S := \mathsf{Res}(f)$, and it thus becomes a less-powerful attacker, i.e., an impersonator. Alternatively, it can also construct the knowledge of f via massive querying, so that later on it can plausibly impersonate S to C, which could however be very costly.

Definition 5.4. *Let $\langle \mathcal{Q}, \mathcal{R}, F \rangle$ be a query-response space, and $\langle \mathsf{Qry}, \mathsf{Res} \rangle$ be a single-round protocol implementing it. For each $q \in \mathcal{Q}$ and each $f \in F$ let $C := \mathsf{Qry}(q)$ and $S := \mathsf{Res}(f)$ and Adv be a PPT algorithm. Denote by* transcript *the communication transcript resulted from the execution of C, S and Adv. Then, Adv is said to be a successful* online man-in-the-middle *(oMitM) attacker if there exist $(C, \mathsf{Adv}, \overline{q}, s_q, t_q)$, $(\mathsf{Adv}, C, r_{\overline{q}}, s_r, t_r)$, $(\mathsf{Adv}, S, q', s_{q'}, t_{q'})$, $(S, \mathsf{Adv}, r', s_{r'}, t_{r'}) \in$ transcript such that*

$$t_q < s_{q'} < t_{r'} < s_r \wedge \overline{q} = q \tag{5.2}$$

where $\overline{q}$ and $r_{\overline{q}}$ denote a query made by C and the corresponding response it received, respectively, and q' and r' denote a query received by S and the corresponding response from S, respectively.

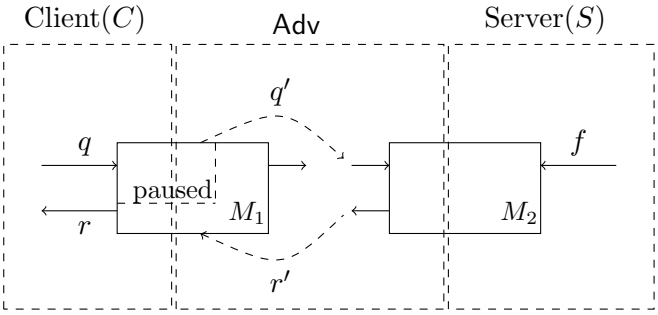


Figure 5.3: MitM attacks as machine execution

In this definition, the notion of adversary is formalised for each pair of query q and function f, since the adversary may discretely choose its behaviour based on the received query and/or the perceived response function. Also, because C, S, and Adv are PPT algorithms, it is possible that (5.2) only holds with some probability p. In that case, we consider that Adv succeeds with probability p. Note also that the definition inherently assumes that Adv has complete knowledge of all messages during the communication. This is a stricter adversary model than cryptography that allows partial knowledge. However, we later show that this assumption is reasonable by making sure, using several cryptographic techniques, that the adversary can intercept/capture either all messages, or none at all.

5.3.2 Solution Motivation and Overview

Our approach to discouraging oMitM attacks stems straightforwardly from Definition 5.4. Indeed, the protocol execution of $C := \mathsf{Qry}(q)$ and $S := \mathsf{Res}(f)$ implies that Adv would need to capture q, and then it must be able to make a query q' to S for $r' = f(q')$. Therefore, if it is possible to make either or both of these two steps costly, there is a potential that an adversary would refrain from attacking.

To visualise this approach, we model an oMitM attack as two concurrent executions of the query-response protocol, one between the client C and Adv, and the other between Adv and the server S, as can be seen in Figure 5.3. Each execution, including the computation at both ends and the communication, can be thought of as inner operation within a (Turing) machine. For convenience we name these machines M_1 and M_2 as in the figure. In this case, M_1 takes as input a query $q \in \mathcal{Q}$, for instance from a user. It outputs a response r to the user, as well as some query representing what Adv received from C. Meanwhile, M_2 takes as input some queries from Adv and a response function $f \in F$, and outputs corresponding responses to Adv.

As the first step of an oMitM attack, Adv must be able to query the server with a

request q' related to q, e.g., $q' = q$. The simplest way to do so is for Adv to honestly follow Res, so that after termination M_1 would correctly produce q, which can then be used to construct q'. However, the termination of M_1 also produces a response r, that is, the client has already received a result for its query. Evidently, such response was not influenced by any communication with the server, and thus the attack is not a valid oMitM attack. Alternatively, Adv may pause the operation of M_1 at some point and make a copy of M_1 along with its current states (memory, registers, etc.), called M_1'. The adversary then resumes M_1' and waits for it to output q, which should be identical to the eventual output of M_1. This would give Adv a chance to perform relevant inquiries to the server before a response is produced to the client. Nevertheless, this attack method only works if Adv has full knowledge of the circuitry of M_1, which involves the PPT algorithms implemented at both the client and Adv. Unfortunately, this is not always guaranteed because the adversary may not know the realisation of the probabilistic nature of the client. In fact, later on in our solution we actually introduce randomness at the client side to prevent this attack from happening.

The remaining option for Adv is to pause M_1, then examines its inner states to extract information about q to produce a related query q'. After getting a response r' for q', it can then modify M_1 so that on termination the output response r is related to r'. However, there is again an obstacle for Adv, which is the potential difficulty in obtaining information about q while examining M_1. This is key to our solution. In a nutshell, we exploit the fact that because there is a lack of obtainable information about q, the adversary must make many different queries to the server in the hope that at least one of them relates to q. By introducing a cost of making queries, the attack may be discouraged if the total cost of querying exceeds the attack benefit. Note however that the querying cost per query also applies to the client, but since the client knows exactly its query, the total cost is much less.

Our solution involves several rounds of querying in the presence of Adv, an example of which is found in Figure 5.4. In particular, the client picks, in addition to the desired query q, a number of other random queries, so as to have a total, say n queries, one for each round. The client then shuffles this set of queries randomly, denoted by $\boldsymbol{q}$. Next, given the shuffled order, the queries are made one-by-one to the server S , which in fact are intercepted by Adv. For each query, Adv must decide whether to perform the oMitM attack, and in overall it is only successful if the oMitM attack is carried on the round where the actual query q is made. Meanwhile, to perform the oMitM attack on each query, the adversary must execute the entire protocol[1] with the actual server. In order to guarantee a successful oMitM attack, the number of queries (e.g. n^2) made by Adv is thus much more than that made by the client (e.g. n). Any attempt to

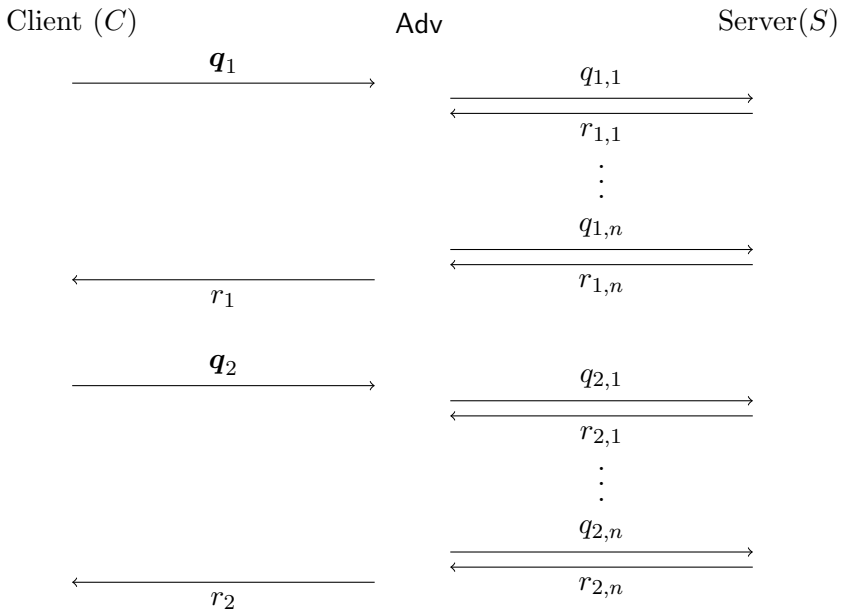


Figure 5.4: Example query-response communication with $n = 2$ rounds

lessen the querying cost would also reduce the adversary's chance of success. When the cost reaches zero (no query), the adversary becomes at most an impersonator (with certainty).

5.4 Defining Security Game

We recall from previous sections that strict security for unauthenticated communication cannot be achieved, i.e., against all efficient adversaries. Therefore we seek the notion of rational security, i.e., safety against rational adversaries who would only attack if there is positive payoff in doing so. To study such possibility, we model the communication as a strategic interaction, i.e., a game among three players: the querier (e.g. user), the responder (e.g. server), and the adversary. Our goal is to design a communication mechanism that restricts the possible actions of the players in the game, so that there would exist an equilibrium point in which the adversary has no benefit in becoming an oMitM attacker.

We present our result in the reverse order. In this section we describe the desirable game which we want the players to play. The next section focuses on showing the existence of equilibrium points. After that we show how players can be forced to play this game via a cryptographic protocol. Indeed, we use cryptographic techniques to

[1]Later on we show, using cryptographic commitments, that the adversary must execute the entire protocol with the server in order to receive any response from the server.

ensure that the adversary cannot perform outstanding behaviours, such as capturing the content of query at early stages, or partially performing an otherwise atomic action specified by the game.

5.4.1 Specifying Environment Parameters

Before formalising the game, we first need to identify assumptions about the operating environment that would help us construct players' utility functions. Our first assumption is about the gain of attacker and loss/damage on the client and server:

Assumption 1 (Transferability)**.** *Let $L_C < 0$ and $L_S < 0$ be the losses of the client and the server compared to a normal attack-free communication, then $G_{\mathsf{Adv}} = -(L_C + L_S)$ is the benefit to the adversary.*

Transferable/zero-sum utility is a natural assumption often used in game-theoretic analysis, which is reasonable as it signifies the fact that benefit neither vanishes nor expands, but is simply transferred from one hand to another. Our argument for using zero-sum gain/loss is likewise. Indeed, assume otherwise that $G_{\mathsf{Adv}} > -(L_C + L_S)$, then, working under the assumption that there exists a "market" for such transactions, the victims (client and server) can at least raise the potential value of the loss by having an option to "sell" the attack to the adversary at price G_{Adv}. In other words, the victim would agree to let the attack happen, or even carry the attack against itself where possible, so as to receive a "fee" of G_{Adv} from the adversary. If the adversary is able to attack without this agreement, the victim inherently loses this amount. On the other hand, having $G_{\mathsf{Adv}} < -(L_C + L_S)$ would undermine the determination of the adversary, and thus is a bad assumption. This concludes our choice of transferability.

For the sake of simplicity, we also assign a single value of damage for each attack type, i.e., impersonation and man-in-the-middle, instead of letting it to be influenced by how the attack is carried out. In reality, the actual damage of an attack also depends on, for example which response is returned to the client, or how much the adversary learns about the query. However, in this case we assume either the average value or the maximum damage, which may be taken from statistics. This can be summarised as below, as well as in Table 5.1.

Assumption 2 (Simplified loss)**.** *Denote by $L_C^{mitm} < 0$ (resp. $L_S^{mitm} < 0$) the loss to the client (resp. server) when Definition 5.4 is satisfied. Denote by $L_C^{imp} < 0$ (resp. $L_S^{imp} < 0$) the utility contribution to the client (resp. server) otherwise.*

Our next argument is about the relation among these losses. For simplicity we assume the same ordering of losses for both the client and the server across all situations,

	Client(C)	Server(S)	Attacker(Adv)
oMitM (Definition 5.4)	L_C^{mitm}	L_S^{mitm}	$G_{\mathsf{Adv}}^{mitm} = -(L_C^{mitm} + L_S^{mitm})$
Impersonation	L_C^{imp}	L_S^{imp}	$G_{\mathsf{Adv}}^{imp} = -(L_C^{imp} + L_S^{imp})$
No attack	0	0	0

Table 5.1: Table of losses in different scenarios.

i.e., for events $A \neq B$, $L_C^A > L_C^B$ implies $L_S^A > L_S^B$ (and vice versa), and similarly for $<$ and $=$ relations. We also assume that $L^{mitm} < L^{imp}$, as otherwise it would be the case that $G_{\mathsf{Adv}}^{mitm} \leq G_{\mathsf{Adv}}^{imp}$, and thus the adversary has no incentive in becoming an oMitM adversary. We omit this case as there is no need for a solution against such adversary.

Assumption 3 (Utility ordering)**.** *We assume the following regarding losses of the client and the server:* $L_C^{mitm} < L_C^{imp}$ *and* $L_S^{mitm} < L_S^{imp}$.

Finally, we introduce the most important component of our game model, which is the cost of querying c, as it is our mean of discouraging oMitM attacks. In association with the querying cost, there is also a cost of processing a query, denoted $\tau(c)$, which we assume to be increasing but negligible, which implies that although the server/responder would prefer less processing, but that is of least importance in its decision-making process. In reality, querying/processing cost can be introduced in several ways, for example as below:

- **Efforts**: the cost may stem from a requirement that before making a query, the querier must perform some expensive tasks. Here the processing cost $\tau(c)$ is the cost of verifying that the task is carried out properly. Expensive tasks are normally facilitated using *Proof-of-Work* (PoW) mechanisms [28], which involves solving a freshly generated puzzle with adjustable difficulty. PoW offers different forms of cost, such as memory [48], network [1], and CPU (e.g., DDH [145], hash inversion [75], factoring [49]).

- **Time**: in time-critical environments, cost may be introduced via mandatory waiting time before a query can be made. Again, $\tau(c)$ comes from the fact that the responder also has to wait before making a response.

- **Monetary**: it might be possible that a querier must pay to make a query. In compatible with our model, we assume that the payment is not made to the responder, but to some non-player party, such as the channel operator, e.g., the ISP. Likewise $\tau(c)$ is shaped by the channel operator's policy. We also assume

that the amount of payment is agreed between the querier and the responder, whereas the bounds on such value is designated by the channel operator.

Assumption 4 (Querying cost)**.** *Denote by* $c_{\min} \geq 0$ *and* $c_{\max} \geq c_{\min} \geq 0$ *the minimum and maximum possible costs for sending a query and receiving a response. The cost is not transferable, that is, it does not come from one player to another. Let* $c \in [c_{\min}, c_{\max}]$ *be the actual querying cost, then the cost of processing a query is* $\tau(c)$*, which is an increasing but negligible function in* c*.*

5.4.2 The High-Level Protocol

In order to specify the communication game, we first need to identify the actions/strategies of involved players. In a nutshell, the game is informally described as follows: the client and server will choose how to communicate, and the adversary will choose how to attack. In this case we are not interested in the possibility that a party can abstain from the communication process, and instead assume that they are always willing to participate. This is because our main focus is to demonstrate that it is possible to discourage oMitM attacks over impersonation, even though the former is more beneficial to the adversary, rather than showing what exactly players should do in reality.

As for the adversary, cryptographic literature imposes no boundary on its possible strategies, apart from being probabilistically polynomial-time (PPT). On the contrary, we employ restrictions on the strategies of the client and the server, mainly by forcing them to follow some communication protocols. There are a few reasons in favour of these restrictions. Firstly, choices made by the client and the server in an actual communication are often preprogrammed into communication software, and thus they are only able to act in certain ways that the software allows. Secondly, having unrestricted actions may result in an optimal choice that is difficult to implement, for example it may require sophisticated cooperation between human and machines. The last reason is to avoid complication of game analysis, as for example the client could have strategies involving out-of-band channels which are difficult to model. On the other hand, we try not to restrict their strategies more than necessary, since that would render the mechanism inflexible, and hence reduce its applicability.

We specify strategies for the client and the server as a set of protocols they may choose in order to communicate with each other. Our protocols are aligned with the solution overview captured in Figure 5.4. We start with a query-response space $\langle \mathcal{Q}, \mathcal{R}, F \rangle$, and let $n_{\max} \leq \lfloor \sqrt{|\mathcal{Q}|} \rfloor$. Our sets of protocols for Qry and Res implementing $\langle \mathcal{Q}, \mathcal{R}, F \rangle$ are given in Figure 5.6 and Figure 5.7, respectively. The protocol uses extra machines,

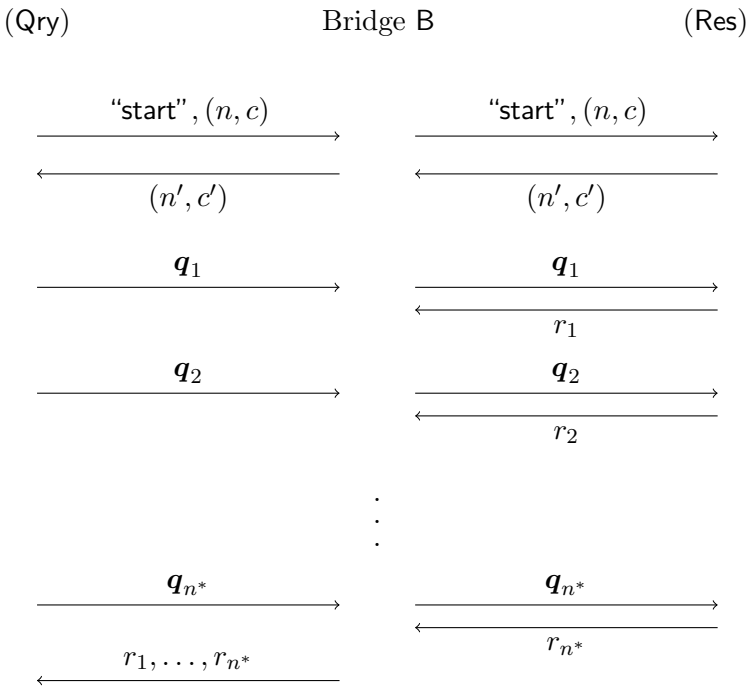


Figure 5.5: (Qry, Res) operation with bridge B

which we call *store-then-forward* bridges, as can be seen in Figure 5.5. The involvement of the adversary requires two bridges to facilitate protocol executions between the adversary and the two endpoints. Although these machines do not exist in reality, they can be implemented using cryptographic commitments, which we show later. Nevertheless, the components of the (Qry, Res) protocol set can be explained as follows:

- $\perp$: a message the does not belong to $\mathcal{Q} \cup \mathcal{R}$.
- $\mathsf{Send}^{dst}(m)$: a subroutine that sends a message m to the destination dst. When it is invoked by src, then m is added to the tail of a queue at dst allocated for messages from src.
- $\mathsf{Receive}^{src}(m)$: a subroutine that receives a message m from the source src. When invoked at dst, it takes a message m out from the head of its local queue allocated for messages from src. If the queue is empty, it waits for a message to come, and timeouts after reasonable waiting time.
- $\mathsf{Send}_c^{dst}(m)$: similar to $\mathsf{Send}^{dst}(m)$, but incurs a cost c to complete.
- $\mathsf{Receive}_c^{src}(m)$: similar to $\mathsf{Receive}^{src}(m)$, but its execution at dst implies a cost c to the immediate sender who executed $\mathsf{Send}_c^{dst}(m)$. If it fails, then m is set as $\perp$.

- $\mathsf{QueryGen}(q, n, r, \mathcal{D})$: given $\mathcal{D}$ as a prior probability distribution of q over $\mathcal{Q}$, selects an n-tuple $\boldsymbol{q}$ of distinct queries in $\mathcal{Q}$ such that $\boldsymbol{q}_r = q$, and that gives no information about r. We provide a construction for this in Proposition 5.2.
- B_1: a machine connected to the client C and Adv. Informally, it serves as a "smart" bridge between Adv and C. Normally messages are simply forwarded in between, with the exception of responses from Adv which will be accumulated and stored in B_1 and only get forwarded to C after all the rounds of querying are completed. For this reason, B_1 is called a *store-then-forward* bridge.
- B_2: similar to B_1, but bridges the adversary Adv and the server S.
- All above subroutines never fail, but only timeout after reasonable waiting time, the effect of which we assume to not influence any player's utility. If a timeout occurs, then the corresponding algorithm is terminated. Note however that the bridges never timeout.

The correctness of $(\mathsf{Qry}, \mathsf{Res})$ is given below:

Proposition 5.1. *For any valid choices of n, c, r and n', c' and any query-response space $\langle \mathcal{Q}, \mathcal{R}, F \rangle$, $(\mathsf{Qry}_{n,c,r}, \mathsf{Res}_{n',c'})$ is a single-round protocol implementing $\langle \mathcal{Q}, \mathcal{R}, F \rangle$.*

Proof. Given that there is no adversary, then the protocol involves $C := \mathsf{Qry}_{n,c,r}(q)$, $S := \mathsf{Qry}_{n',c'}(f)$, and bridge B, for some $q \in \mathcal{Q}$ and $f \in F$. The protocol starts with the client C sending a message "start" to initiate the conversation. The server and the client then exchange their round complexity parameters, i.e., (n', c') and (n, c), respectively. The round complexity is agreed to be the maximum of the two choices: $n^* = \max(n, n')$ and $c^* = \max(c, c')$, where n^* is the number of queries/rounds the client will make, and c^* is the cost the client incurs in each querying round. This is followed by a sequence of rounds, in each of which C make a query $\boldsymbol{q}_i$ and the server provides a response r_i, which is held at $\mathbf{B}$. After n^* rounds, all responses are flushed back to C, thus satisfying Definition 5.3. □

Proposition 5.2. *Let $\langle \mathcal{Q}, \mathcal{R}, F \rangle$ be a query-response space, and $\mathcal{D}$ be a probability distribution over $\mathcal{Q}$. Define $\mathsf{QueryGen}(q, n, r, \mathcal{D})$ as follows:*

$$
\begin{aligned}
&\mathcal{Q}_q \leftarrow \{q' \in \mathcal{Q} \mid \Pr_{\mathcal{D}}[q'] = \Pr_{\mathcal{D}}[q]\} \\
&\textbf{\textit{if}}\ |\mathcal{Q}_q| < n\ \textbf{\textit{then return}}\ \perp \\
&s \leftarrow \{q\} \\
&\textbf{\textit{for}}\ i = 1\ \textbf{\textit{to}}\ n - 1
\end{aligned}
$$

Protocol $\mathsf{Qry}_{n,c,r}(q)$

Parameters: $n \in [n_{\max}]$, $c \in [c_{\min}, c_{\max}]$, $r : [n_{\max}] \times [c_{\min}, c_{\max}] \to [n_{\max}]$ s.t. $r(n', c) \leq \max(n', n)$ $\forall (n', c) \in [n_{\max}] \times [c_{\min}, c_{\max}]$.

Channels: peer.

Input: A query $q \in \mathcal{Q}$ and a probability distribution $\mathcal{D}$ over $\mathcal{Q}$.

Protocol.

$\mathsf{Send}^{\mathsf{peer}}(\text{“start”})$
$\mathsf{Send}^{\mathsf{peer}}(n, c)$
$\mathsf{Receive}^{\mathsf{peer}}(n', c')$
if $n' \notin [n_{\max}] \vee c \notin [c_{\min}, c_{\max}]$ **then return** $\perp$
$n^* \leftarrow \max(n, n')$
$c^* \leftarrow \max(c, c')$
$\boldsymbol{q} \leftarrow \mathsf{QueryGen}(q, n^*, r(n', c'), \mathcal{D})$
if $n'' \neq n^*$ **then return** $\perp$
for i **from** 1 **to** n^*
  $\mathsf{Send}^{\mathsf{peer}}_{c^*}(\boldsymbol{q}_i)$
  $\mathsf{Receive}^{\mathsf{peer}}(\mathsf{sig})$
  if $sig \neq$ “done” **then return** $\perp$
for i **from** 1 **to** n^*
  $\mathsf{Receive}^{\mathsf{peer}}(R_i)$
if $\perp \in \{R_1, \ldots, R_{n^*}\}$ **then return** $\perp$
return $R_{r(n', c')}$

Figure 5.6: The set of Qry client strategies over the choices of n, c, r.

Protocol $\mathsf{Res}_{n',c'}(f)$

Parameters: $n' \in [n_{\max}]$, $c' \in [c_{\min}, c_{\max}]$.

Channels: peer.

Input: A response function $f \in F$.

Protocol.

$\mathsf{Receive}^{\mathsf{peer}}(sig)$
while $sig =$ "start" **do**
  $\mathsf{Receive}^{\mathsf{peer}}(n, c)$
  $\mathsf{Send}^{\mathsf{peer}}(n', c')$
  if $n' \notin [n_{\max}] \vee c \notin [c_{\min}, c_{\max}]$ **then continue**
  $n^* \leftarrow \max(n, n')$
  $c^* \leftarrow \max(c, c')$
  for i **from** 1 **to** n^*
    $\mathsf{Receive}^{\mathsf{peer}}_{c^*}(q)$
    if $q \in \mathcal{Q}$ **then** $\mathsf{Send}^{\mathsf{peer}}(f(q))$
      else $\mathsf{Send}^{\mathsf{peer}}(\perp)$
    $\mathsf{Send}^{\mathsf{peer}}(\text{"done"})$
  $\mathsf{Receive}^{\mathsf{peer}}(sig)$

Figure 5.7: The set of Res server strategies over the choices of n' and c'.

Protocol B

Channels: src and dst.

Protocol:

while true do
  $\mathsf{Receive}^{src}(sig)$
  $\mathsf{Send}^{dst}(sig)$
  $\mathsf{Receive}^{src}(n, c)$
  $\mathsf{Send}^{dst}(n, c)$
  $\mathsf{Receive}^{dst}(n', c')$
  $\mathsf{Send}^{src}(n', c')$
  if $sig \neq$ "start" $\vee\, n, n' \notin [n_{\max}] \vee c, c' \notin [c_{\min}, c_{\max}]$ **then**
    continue
  $n^* \leftarrow \max(n, n')$
  for i **from** 1 **to** n^*
    $\mathsf{Receive}^{dst}(R_i)$
    $\mathsf{Receive}^{dst}(sig)$
    $\mathsf{Send}^{src}(sig)$
  for i **from** 1 **to** n^*
    $\mathsf{Send}^{dst}(R_i)$

Figure 5.8: Operation of store-then-forward bridge B_1 (resp. B_2) where $(src, dst) = (\mathsf{Qry}, \mathsf{Adv})$ (resp. $(\mathsf{Adv}, \mathsf{Res})$).

$$
\begin{aligned}
& q_i \leftarrow_{\$} \mathcal{Q}_q \setminus s \\
& s \leftarrow s \cup \{q_i\} \\
& q_n \leftarrow q_r \\
& q_r \leftarrow q \\
& \textbf{\textit{return}}\ (q_1, \ldots, q_n).
\end{aligned}
$$

Then the output of QueryGen *gives no information about* r*. In other words, for all* $n \in \mathbb{N}^+$*, all distinguisher* M *and* $r_0, r_1 \in [n]$ *we have*

$$
\begin{aligned}
&\Pr[q \leftarrow_{\mathcal{D}} \mathcal{Q}; \boldsymbol{q} \leftarrow \mathsf{QueryGen}(q, n, r_0, \mathcal{D}) : M(\boldsymbol{q}, n) = 1] \\
=&\Pr[q \leftarrow_{\mathcal{D}} \mathcal{Q}; \boldsymbol{q} \leftarrow \mathsf{QueryGen}(q, n, r_1, \mathcal{D}) : M(\boldsymbol{q}, n) = 1]
\end{aligned}
$$

Proof. To prove this result, we only need to show that QueryGen produces probabilistically the same output regardless of the value of r, and since M receives the same input, it must give the same output. Indeed, we first notice that QueryGen outputs $=\perp$ with the same probability for both r_0, and r_1. This is because the condition for outputting $\perp$ is $n > |\mathcal{Q}_q|$ which is independent of r. Assume otherwise that QueryGen always outputs a valid query tuple, then it is easy to see from the algorithm above that QueryGen is a simple random sampling without replacement of equal probabilities over the set $\mathcal{Q}_q$. This means that any permutation of the sample will have the same probability distribution. On the other hand, $\mathsf{QueryGen}(q, n, r_0, \mathcal{D})$ can be converted to $\mathsf{QueryGen}(q, n, r_1, \mathcal{D})$ by swapping q_n with q_{r_0}, then q_{r_1} with q_n, which results in a permutation. Thus any distinguisher would have the same view in both cases. □

5.4.3 Game Formalisation

In the previous subsection we declare the strategies of players. Our next concern is on the type of game that appropriately captures the players' interactions. For this we first notice that our oMitM attacks are defined over choices of query q and response function f. In addition, our assumptions on losses (transferability, simplified losses) indicate that players' utilities are influenced by the type of attack, i.e., Definition 5.4, which is subject to the value of the query q. Since the adversary and the server do not know q before taking an action, they thus have incomplete knowledge of their utility. This suggests us to consider using Bayesian games in the analysis.

We start with the standard definition of Bayesian game in Definition 1.6, i.e., a tuple $\langle N, \Omega, \langle A_i, T_i, C_i, \tau_i, p_i, u_i \rangle_{i \in N} \rangle$. Our game is played between the client, the server, and the adversary, meaning that $N = \{C, S, \mathsf{Adv}\}$. Here the state of nature essentially

captures the unknown parameter, i.e., q, and hence with respect to a query-response space $\langle \mathcal{Q}, \mathcal{R}, F \rangle$ we set $\Omega = \mathcal{Q}$. As a proof of concept, we assume that the query is picked uniformly randomly from $\mathcal{Q}$. In other words, $p_C(\omega) = p_{\mathsf{Adv}}(\omega) = p_S(\omega) = 1/|\mathcal{Q}|$ for all $\omega \in \Omega$.

Next, A_i captures available actions for players, in which the strategies for the client and the server are described in Figure 5.6 and Figure 5.7, respectively. Meanwhile, the adversary strategy set is the set of all possible PPT algorithms. For consistency, we then have the client's type and signal function as $T_C = \mathcal{Q}$ and $\tau_C(\mathfrak{q}) = \mathfrak{q}$ for all $\mathfrak{q} \in \Omega$. Likewise, we also have $T_S = T_{\mathsf{Adv}} = \{0\}$ and $\tau_S(\mathfrak{q}) = \tau_{\mathsf{Adv}}(\mathfrak{q}) = 0$ for all $\mathfrak{q} \in \Omega$. As for type-to-actions mappings C_i, we notice that since players' strategies are in the form of Turing machines, which accept all possible inputs, therefore $C_i(t_i) = A_i$ for all $i \in N$ and all $t_i \in T_i$.

What is left to identify is $u_i : \Omega \times A \to \mathbb{R}$, i.e., the utility function for each player, where $A = A_C \times A_S \times A_{\mathsf{Adv}}$. Essentially, the utility for each player is composed of two parts: its gain/loss as the result of an attack, and the cost for carrying out its action. Following Definition 5.4, the first part can inherently be inferred from the communication transcript of $C := \mathsf{Qry}(q)$, $S := \mathsf{Res}(f)$ and Adv, where $(\mathsf{Qry}, \mathsf{Res}, \mathsf{Adv}) \in A$. Note that even though B_1 and B_2 are network entities, but since they honestly bridge messages (with delays), we do not consider them in the communication transcript. Meanwhile, the cost of action is essentially the number of queries multiplied by the cost of making query (resp. processing query) for the querier (resp. responder). For the convenience of players' utilities, for each $(\omega, a) = (q, (\mathsf{Qry}, \mathsf{Res}, \mathsf{Adv})) \in \Omega \times A$ we define the following:

$$L_C(\omega, a) = \begin{cases} L_C^{mitm} & \text{if Definition 5.4 is satisfied, or} \\ L_C^{imp} & \text{otherwise.} \end{cases}$$

$$L_S(\omega, a) = \begin{cases} L_S^{mitm} & \text{if Definition 5.4 is satisfied, or} \\ L_S^{imp} & \text{otherwise.} \end{cases}$$

$\boldsymbol{q}(\omega, a)$: the set of queries received by S

$c_C(\omega, a)$: cost value c^* in the description of Qry

$c_S(\omega, a, \mathfrak{q})$: cost value c in $\mathsf{Receive}_c(\mathfrak{q})$ in the description of Res

$n_C(\omega, a)$: number of times Qry invokes Send_{c^*}

Then, players' utilities can be constructed as follows:

$$u_C(\omega, a) = \mathbb{E}\left[L_C(\omega, a) - c_C(\omega, a) * n_C(\omega, a)\right] \tag{5.3}$$

$$u_{\mathsf{Adv}}(\omega, a) = \mathbb{E}\left[-(L_C(\omega, a) + L_S(\omega, a)) - \sum_{\mathfrak{q} \in \boldsymbol{q}(\omega, a)} c_S(\omega, a, \mathfrak{q})\right] \tag{5.4}$$

$$u_S(\omega, a) = \mathbb{E}\left[L_S(\omega, a) - \sum_{\mathfrak{q} \in \boldsymbol{q}(\omega, a)} \tau\left(c_S(\omega, a, \mathfrak{q})\right)\right] \tag{5.5}$$

We notice that in this case the cost of querying for the client is the same for each query, as can be straightforwardly inferred from Figure 5.6. In contrast, at the server side, there is a chance for the querying cost to change after each of the **while** loop, hence the cost must be calculated per query, which explains the summation $\sum$ over $\boldsymbol{q}(\omega, a)$. Also, the utilities are of expected values due to the probabilistic nature of the adversary. For convenience, the game is properly defined as follows:

Definition 5.5. *Let $\langle \mathcal{Q}, \mathcal{R}, F \rangle$ be a query response space. Let $\{\mathsf{Qry}\}$ and $\{\mathsf{Res}\}$ be strategy spaces defined as in Figure 5.6 and Figure 5.7, respectively. With respect to above components, an **unauthenticated communication game** is a Baysian game $\langle N, \Omega, \langle A_i, T_i, C_i, \tau_i, p_i, u_i \rangle_{i \in N} \rangle$ such that:*

- $N = \{C, S, \mathsf{Adv}\}$,
- $\Omega = \mathcal{Q}$,
- $A_C = \{\mathsf{Qry}\}$, $A_S = \{\mathsf{Res}\}$, *and* A_{Adv} *is the set of all PPT algorithms,*
- $T_C = \mathcal{Q}$, $T_S = \{0\}$, *and* $T_{\mathsf{Adv}} = \{0\}$,
- $C_i = A_i$,
- $\tau_C(\omega) = \mathfrak{q}$, $\tau_S(\omega) = 0$, *and* $\tau_{\mathsf{Adv}}(\omega) = 0$ *for all* $\omega = \mathfrak{q} \in \Omega$,
- $p_C(\omega) = p_S(\omega) = p_{\mathsf{Adv}}(\omega) = 1/|\mathcal{Q}|$ *for all* $\omega \in \Omega$,
- u_i *are as in* (5.3), (5.4), (5.5).

5.5 Game Analysis

In this section we analyse the previously specified game and construct equilibria in which the adversary cannot successfully perform an oMitM attack. In other words, the outcome of the equilibrium is that with certainty Definition 5.4 is not satisfied. To facilitate equilibria computation, we first provide categorisation of the adversary's strategies, along with elimination of weakly dominated strategies, thus making them much more tractable than the set of all possible PPT algorithms.

5.5.1 Simplifying Attack Strategies

In general, the behaviour of the adversary consists of two parts: external communication with different parties, and internal computation, which may happen in parallel or in an arbitrarily mixed order. For the sake of our game analysis, we do not need to study all details about adversary's strategies. This is because the following is true with regards to players' utility functions:

- Players' utilities can be computed solely from the observing the communication among entities.
- Players utilities do not depend on the values of query/response messages, only on their order appearance.

Hence, we omit the adversary's internal computation, and only mention it briefly in our analysis. In the following we construct the general algorithm that captures the adversary's possible communication, based mainly on its protocol execution with the client $C := \mathsf{Qry}(q)$, for two reasons. Firstly, since the adversary cannot actively start a protocol execution with C, the behaviour of C is fixed and well-known, which is ideal for building a framework of the adversary's strategies. Secondly, the timeline of this execution would help us detect the type of attack that may occur. In Figure 5.9 we construct the general structure of Adv's communication, which is supported by the following result:

Lemma 5.1. *Let $\langle \mathcal{Q}, \mathcal{R}, F \rangle$ be a query-response space. For some $c_{\min}, c_{\max} > 0$, let $\{\mathsf{Qry}\}$ and $\{\mathsf{Res}\}$ be defined according to Figure 5.6 and Figure 5.7, respectively. Let $G = \langle N, \Omega, \langle A_i, T_i, C_i, \tau_i, p_i, u_i \rangle_{i \in N} \rangle$ be an unauthenticated communication game with respect to above components. Let $A_{\mathsf{Adv}}^{\mathsf{reduced}}$ be the set of PPT algorithms such that their communication framework are as in Figure 5.9 for all states of nature $\omega \in \Omega$ and all $a_{-i} \in A_{-i}$. Then A_{Adv} reduces to $A_{\mathsf{Adv}}^{\mathsf{reduced}}$ via elimination of weakly dominated strategies and/or equivalent strategies*[2].

Proof. Our proof has three steps. In the first step we show that the adversary Adv's communication with the client C must be as specified. Next, we explain why Adv only communicates with the server S at certain points as in Figure 5.9. Finally, we prove that Adv should communicate with S in a way suggested by Figure 5.9.

For the first part of the proof, we notice that the specification of the adversary Adv's communication with C in Figure 5.9 resembles that in Figure 5.7. Suppose otherwise

[2]Two strategies s and s' of player i are equivalent if $\mathbb{E}_\omega u_i(\omega, s, s_{-i}) = \mathbb{E}_\omega u_i(\omega, s', s_{-i})$ for all $s_{-i} \in A_{-i}$.

Protocol Adv

Channels: C, S.

Protocol:

$\mathsf{Receive}^C(\text{“start”})$
$\mathsf{Receive}^C(n_C, c_C)$
$\mathsf{Send}^S(\text{“start”})$
$\mathsf{Send}^S(n', c')$ where $n' \notin [n_{\max}] \vee c' \notin [c_{\min}, c_{\max}]$
$\mathsf{Receive}^S(n_S, c_S)$
$\mathsf{Send}^C(n_{\mathsf{Adv}}, c_{\mathsf{Adv}})$ for $(n_{\mathsf{Adv}}, c_{\mathsf{Adv}}) \in P$
$n^* \leftarrow \max(n_C, n_{\mathsf{Adv}})$
for i **from** 1 **to** n^*
  $\mathsf{Receive}^C(\boldsymbol{q}_i)$
  Execute Qry to S with $(n', c') = (1, c_{\min})$ (Optional)
  $\mathsf{Send}^C(r_i)$
  $\mathsf{Send}^C(\text{“done”})$

Figure 5.9: Generic adversary's strategies where optional steps can be skipped.

that Adv either omit some of these steps, or arrange them in a different order, or both. We discuss omission of steps as below:

- $\mathsf{Receive}^C(\text{“start”})$ and $\mathsf{Receive}^C(n_C, c_C)$: skipping these steps has the same effect as executing them and then ignoring the received messages.
- $\mathsf{Send}^C(n_{\mathsf{Adv}}, c_{\mathsf{Adv}})$, $\mathsf{Send}^C(r_i)$, and $\mathsf{Send}^C(\text{“done”})$: skipping these steps will cause the client C to timeout and acknowledges no response for any of its queries, and thus together failing Definition 5.4. This is the worst case for the adversary since executing these steps incur no cost. Thus it is in Adv's best interest to execute these steps.
- $\mathsf{Receive}^C(\boldsymbol{q}_i)$: the reasons for executing this step are the same for previous two cases. In other words, skipping it is the same as executing it and then ignoring $\boldsymbol{q}_i$, and also reduces the chance that Definition 5.4 is satisfied.

Consider the option of shuffling the above steps in a different order. Assume that this does not cause the client C to timeout or terminate, then the Adv receives the same set of messages from C, and hence gains no additional benefit from their contents. The only way shuffling could affect the adversary's utility is via the order of appearance of $\mathsf{Receive}^C(\boldsymbol{q}_i)$ and $\mathsf{Send}^C(r_i)$, as they relate to Definition 5.4. However, if $\mathsf{Receive}^C(\boldsymbol{q}_i)$ occurs after $\mathsf{Send}^C(r_i)$, this means that for messaging events $(C, \mathsf{Adv}, \boldsymbol{q}_i, s_q, t_q)$ and $(\mathsf{Adv}, C, r_i, s_r, t_r)$ we have $s_r < t_q$, which clearly does not improve the chance for Definition 5.4 to be satisfied. Thus, shuffling of prescribed steps can be safely eliminated.

Next we show that communication with S should only be made at specific moments relative to Adv's communication with C. Suppose otherwise that communication with S can occur at different stages below, we show that this is fruitless:

- Before $\mathsf{Receive}^C(n_C, c_C)$: this has the same effect as for Adv to wait for $\mathsf{Receive}^C(n_C, c_C)$ to finish and then ignore the received messages while communicating with S.
- After $\mathsf{Send}^C(n_{\mathsf{Adv}}, c_{\mathsf{Adv}})$ and before $\mathsf{Receive}^C(\boldsymbol{q}_1)$: similar to above, and additionally, any communication with S at this stage does not improve the chance of satisfying Definition 5.4, as there is yet any messaging event $(C, \mathsf{Adv}, q, s, t)$.
- After $\mathsf{Send}^C(r_i)$ and before $\mathsf{Receive}^C(\boldsymbol{q}_{i+1})$: this is similar to the previous point, as any communication with S at this stage does not change Adv's communication with C, and also does not fall between any pair of messaging events $(C, \mathsf{Adv}, \boldsymbol{q}_i, s_q, t_q)$ and $(C, \mathsf{Adv}, r_i, s_r, t_r)$.

The final part of the proof is to show that the prescribed communication with the server S is optimal. Following Figure 5.9, the first three steps are $\mathsf{Send}^S(\text{“start”})$,

$\mathsf{Send}^S(n', c')$, and $\mathsf{Receive}^S(n_S, c_S)$, which serve as initialisation of a query-response process with the server. The purpose of this initialisation is to observe (n_S, c_S), which is a server's private information apart from its response function $f \in F$. The value of (n_S, c_S) might be important to Adv in deciding $(n_{\mathsf{Adv}}, c_{\mathsf{Adv}})$ that would be sent back to the client C, as it would contribute to influencing the adversary's utility (5.4). In addition, the use of invalid round parameters $n' \notin [n_{\max}]$ or $c' \notin [c_{\min}, c_{\max}]$ will cause the stateless server S to reset (Figure 5.7), thus creating no effect to any future communication. It is therefore optimal for Adv to execute these steps to learn (n_S, c_S). On the other hand, Adv needs not communicate further with S in order to decide $(n_{\mathsf{Adv}}, c_{\mathsf{Adv}})$, because the best Adv can gain is information about f via the S's responses to queries. However, players' utilities are independent of the value of f, and therefore the adversary needs not learn it at this stage.

The remaining point is to reason why after each $\mathsf{Receive}^C(\boldsymbol{q}_i)$, the adversary should execute a query-response protocol with S at most once, with round parameters (n', c') $= (1, c_{\min})$. Indeed, the idea of communicating with S between $\mathsf{Receive}^C(\boldsymbol{q}_i)$ and $\mathsf{Send}^C(r_i)$ is to ensure that Definition 5.4 is satisfied once $\boldsymbol{q}_i = q$. In other words, this communication must yield a communication transcript containing messaging events $(\mathsf{Adv}, S, q', s_{q'}, t_{q'})$ and $(S, \mathsf{Adv}, r', s_{r'}, t_{r'})$ where r' is the response by S to q'. Because of the feature of B_2 that delays the bridging of responses, Adv must make $n = \max(n_S, n')$ number of queries. To send each query Adv must executing $\mathsf{Send}_c^S(q')$, where $c = \max(c_S, c')$, or otherwise the server's execution of $\mathsf{Receive}_c^{\mathsf{Adv}}(q')$ will fail, causing r' to be set as $\perp$, thus failing Definition 5.4. In overall this incurs a cost $c \cdot n$, which is minimised by setting $(n', c') = (1, c_{\min})$. This process can be conveniently achieved by executing $\mathsf{Qry}_{1,c_{\min},r}$ for arbitrary query-hiding function r. Note also that to satisfy Definition 5.4 the adversary needs not execute Qry more than once. Therefore it is amongst the optimal choices for the adversary to behave a prescribed.

□

The above lemma implies a significant elimination of attack strategies that allow us to parameterise them. Indeed, there are only a few parameters in Figure 5.9 that can be customised by the adversary:

- $n_{\mathsf{Adv}} : P^2 \to [n_{\max}]$, $c_{\mathsf{Adv}} : P^2 \to [c_{\min}, c_{\max}]$: the probabilistic choice of round complexity when negotiating with Qry, decided by Adv after observing (n_C, c_C) and (n_S, c_S).
- $d_i : P^2 \to \{0, 1\}$: the adversary's probabilistic decision whether to query S at round i, where 1 indicates a communication to S and 0 otherwise. Potentially it is a function of several parameters: other players' choices of round complexity,

the client's choice of where (r) to hide q, the client's choice of queries $\boldsymbol{q}$. We however show in Lemma 5.2 that the d_i only needs to depend on other player's round complexity parameters, i.e., (n_C, c_C) and (n_S, c_S). Also, as in the proof of Lemma 5.1, the adversary's utility does not depend on the response function f, and hence it needs not base the choice of d_i on its communication with S.

Lemma 5.2. *Consider an unauthenticated communication game where the adversary's strategy is* $(n_{\mathsf{Adv}}, c_{\mathsf{Adv}}, d_1, \ldots, d_{n_{\max}})$*, then there exists an equivalent strategy* $(n'_{\mathsf{Adv}}, c'_{\mathsf{Adv}}, d'_1, \ldots, d'_{n_{\max}})$ *with probabilistic functions* $d'_i : P^2 \to \{0, 1\}$.

Proof. We first set the probabilistic round complexity parameters $n'_{\mathsf{Adv}} := n_{\mathsf{Adv}}$ and $c'_{\mathsf{Adv}} := c_{\mathsf{Adv}}$. Let (n_C, c_C, r) and (n_S, c_S) be strategies of the client C and the server S, respectively. Then it must be the case that d_i is probabilistically identical for all valid r and i, as otherwise it is easy to construct a distinguisher that violates Proposition 5.2. Denote by $\Pr[d_i = 0 \mid n_C, c_C, n_S, c_S, q]$ the probability that $d_i = 0$ given player's strategies and query q, we construct a strategy such that for all $i \in [n_{\max}]$,

$$\Pr[d'_i = 0 \mid n_C, c_C, n_S, c_S] = \sum_{\mathcal{Q}} \Pr[d_i = 0 \mid n_C, c_C, n_S, c_S, q].$$

Note that such construction is always possible, and it takes as input only (n_C, c_C, n_S, c_S) before outputting the probabilistic decisions of d'_i. Therefore we only need $d'_i : P^2 \to \{0, 1\}$. This mixed strategy yields the same adversary's utility since it leads to the same expected number of queries made by the adversary to the server, i.e.,

$$n_S \sum_{i=1}^{\max(n_{\mathsf{Adv}}, n_C)} \Pr[d'_i = 0 \mid n_C, c_C, n_S, c_S] = n_S \sum_{i=1}^{\max(n_{\mathsf{Adv}}, n_C)} \sum_{\mathcal{Q}} \Pr[d_i = 0 \mid n_C, c_C, n_S, c_S, q] \tag{5.6}$$

as well as the same success probability of satisfying Definition 5.4 which is determined by checking whether $d'_{r(n_{\mathsf{Adv}}, c_{\mathsf{Adv}})} = 1$:

$$\Pr[d'_{r(n_{\mathsf{Adv}}, c_{\mathsf{Adv}})} = 1 \mid n_C, c_C, n_S, c_S] = \sum_{\mathcal{Q}} \Pr[d_{r(n_{\mathsf{Adv}}, c_{\mathsf{Adv}})} = 1 \mid n_C, c_C, n_S, c_S, q]. \tag{5.7}$$

□

Since we assume that the adversary is a PPT algorithm, each attack strategy can thus have a probabilistic choice over n_{Adv}, c_{Adv} and d_i. For convenience of game analysis however, we equivalently consider the deterministic choices as pure attack strategies, and represent probabilistic choices by mixed strategies. The simplification of attack strategies leads to more tractable players' utility functions:

Lemma 5.3. *Consider an unauthenticated communication game where the client's strategy is* (n_C, c_C, r)*, the server's strategy is* (n_S, c_S)*, and the adversary's strategy is categorised by a tuple* $(n_{\mathsf{Adv}}, c_{\mathsf{Adv}}, d_1, \ldots, d_{n_{\max}})$*, then players' utilities are:*

$$u_C(\omega, a) = L_C(\omega, a) - \max(n_C, n)\max(c_C, c) \tag{5.8}$$

$$u_{\mathsf{Adv}}(\omega, a) = -(L_C(\omega, a) + L_S(\omega, a)) - c_S n_S \sum_{i=1}^{\max(n_C, n)} d_i(param) \tag{5.9}$$

$$u_S(\omega, a) = L_S(\omega, a) - \tau(c_S) n_S \sum_{i=1}^{\max(n_C, n)} d_i(param) \tag{5.10}$$

where

$$param = (n_C, c_C, n_S, c_S), n = n_{\mathsf{Adv}}(param), c = c_{\mathsf{Adv}}(param),$$

$$L_C(\omega, a) = \begin{cases} L_C^{mitm} & \text{if } d_{r(n,c)}(param) = 1, \text{ or} \\ L_C^{imp} & \text{otherwise.} \end{cases}, \text{ and,}$$

$$L_S(\omega, a) = \begin{cases} L_S^{mitm} & \text{if } d_{r(n,c)}(param) = 1, \text{ or} \\ L_S^{imp} & \text{otherwise.} \end{cases}.$$

Proof. The proof for this lemma is rather straightforward. We first notice that $r(n, c)$ is the round number of the querying round in which the actual query is made, i.e., $\boldsymbol{q}_{r(n,c)} = q$. Meanwhile, $d_{r(n,c)}(param) = 1$ indicates that within this round, Adv communicates with the server S after $\mathsf{Receive}^C(\boldsymbol{q}_{r(n,c)})$, and before $\mathsf{Send}^C(r_{r(n,c)})$, thus successfully satisfying Definition 5.4 with certainty. Therefore, players' benefit from attack is deterministic as expressed. The querying cost of their utilities are trivial, where $n_S \sum_{i=1}^{\max(n_C, n)} d_i(param)$ is the total number of queries Adv makes to S. □

The above lemma allows us to simplify the notion of unauthenticated communication game, from Bayesian game to a simple strategic-form game. Indeed, players' utilities as expressed in Lemma 5.2 depend on their strategies but the state of nature $\omega \in \Omega$. In addition, players' strategies also do not depend on ω. Therefore we can redefine our game easily in strategic-form.

Definition 5.6. *Let* $\langle \mathcal{Q}, \mathcal{R}, F \rangle$ *be a query response space with some* $\mathsf{Ans} \in F$*, and* $[c_{\min}, c_{\max}]$ *be a cost interval, with* $0 \leq c_{\min} \leq c_{\max}$*. With respect to above components, a* ***refined unauthenticated communication game*** *is a strategic-form game* $\langle N, \{A_i\}_{i \in N}, \{u_i\}_{i \in N} \rangle$ *such that:*

- $N = \{C, S, \mathsf{Adv}\}$,

- $A_C = \{(n_C, c_C, r)\}$, $A_S = \{(n_S, c_S)\}$, *and* $A_{\mathsf{Adv}} = \{(n_{\mathsf{Adv}}, c_{\mathsf{Adv}}, d_1, \ldots, d_{n_{\max}})\}$,
- u_i *are as in* (5.8), (5.9) *and* (5.10).

5.5.2 Finding Good Equilibria

We apply the most intuitive method to find equilibrium for this three-player game. In particular, we fix the strategy of one player, and find an equilibrium of the induced game between the remaining players. This is repeated for every strategy, until we find one that is also a best response against the choices (in equilibrium) of other players. For this we realise that the utilities of the client and the server do not strongly depend on each other's strategy. Therefore, if we choose either one and fix its strategy, then there is a better chance to reuse a two-player equilibrium for a new fixed strategy of that player. In particular, we choose to fix the server's strategy as it has a smaller strategy set, and thus less repetition of equilibrium finding is required. In terms of equilibrium type, our game formulation suggests that we use the notion of "one-shot" Nash equilibrium. However, because the nature of communication is sequential, it is thus more desirable to seek a refinement of such equilibrium, i.e., a perfect Bayesian equilibrium. We achieve this by breaking players' strategies/algorithms into steps and represent the game in extensive form.

By fixing the server's strategy, we are given $a_S = (n_S, c_S)$ for some $(n_S, c_S) \in P$, and the refined unauthenticated communication game is now played between the client (C) and the adversary (Adv). Our goal is to find equilibria in which the adversary would avoid oMitM attacks. In the language of security, this means to look for client strategies that minimise the adversary's information-theoretic advantages. In game theory this would mean to minimise the adversary's maximum possible gain. Our approach is thus similar to finding *minimax* strategy in *zero-sum* games. In fact, our game shares a similar structure to zero-sum games due to the transferability of payoff.

Whilst the game tree is depicted in Figure 5.10, we fix the server's strategy, i.e., $(n_S, c_S) \in A_S$. The game play, following the communication and Definition 5.6 is:

1. The client picks a round complexity parameter (n_C, c_C) and send them out.
2. The adversary, on receipt of (n_C, c_C) and (n_S, c_S), picks a round complexity parameter $(n_{\mathsf{Adv}}, c_{\mathsf{Adv}})$ and send them back.
3. The client decides where to hide its query, and the adversary (simultaneously) decides which of the queries to perform oMitM attacks.

We start by analysing the smallest possible subgame, which basically emerges from each node where (n_C, c_C) and $(n_{\mathsf{Adv}}, c_{\mathsf{Adv}})$ are set. To see this, we recall that $(n_{\mathsf{Adv}}, c_{\mathsf{Adv}})$ is

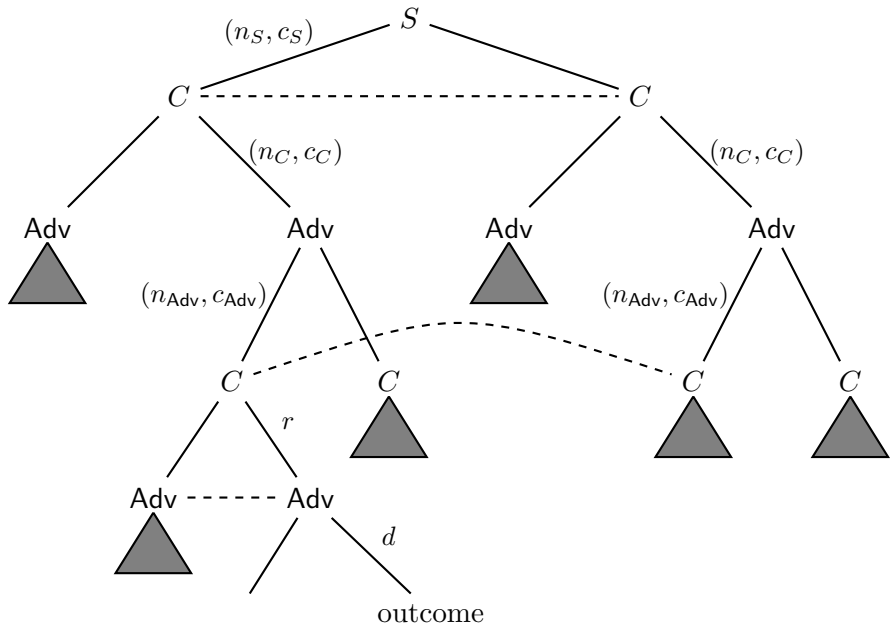


Figure 5.10: Illustration of the game tree, where triangles indicate omitted subgames.

a function over (n_C, c_C) and (n_S, c_S), which means that Adv must have observed C's choice of these values. Likewise, r is a function over $(n_{\mathsf{Adv}}, c_{\mathsf{Adv}})$, and hence C must have observed Adv's choice before deciding where to hide q in the query list. Obviously this can also be explained by looking at the protocol specifications, i.e., players must see each other's choice of round complexity parameters during negotiation before moving on to the querying phase.

Getting back to the subgame, we can easily see that it is actually a "one-shot" game, where informally, the client decides where to hide the actual query q, whilst the adversary decides which of the queries from the client will be queried further to the server in attempting an oMitM attack. Following (5.9), the attack is only successful if q falls within the set of attempted queries. The following result establishes the Nash equilibrium for each of these subgames.

Lemma 5.4. *Let (n_C, c_C), (n_S, c_S) and $(n_{\mathsf{Adv}}, c_{\mathsf{Adv}})$ be given and let $n^* = \max(n_C, n_{\mathsf{Adv}})$. In the induced subgame let a_C denote the client's mixed strategy in which $r \in [n^*]$ is selected uniformly random. Let $G = L_C^{imp} + L_S^{imp} - L_C^{mitm} - L_S^{mitm}$. Denote the adversary's strategy by $d \in \mathbb{Z}_{2^{n^*}}$ such that d_k is the k-th most significant bit of d for all $k \in [n^*]$, i.e., $d_k = (d \gg (n-k)) \wedge 1$. Then:*

- *If and only if $G \leq n_S c_S n^*$ there exists a mixed-strategy Nash equilibrium of the form (a_C, d) for $d = 0b0\ldots0$ (no oMitM attack).*

- *If and only if* $G \geq n_S c_S n^*$ *there exists a mixed-strategy Nash equilibrium of the form* (a_C, d) *for* $d = 0b1 \ldots 1$ *(oMitM attack with certainty).*

Proof. Let $n = n^*$ for convenience of presentation. The subgame involves the client picking a round $r \in [n]$ to hide the query, and the adversary decides which of the rounds to mount oMitM attacks, that is, to query S within those rounds. We denote the client's action r by $i = n - r$, so that $i \in \mathbb{Z}_n$, and the adversary's action d by $j = d$ so that $j \in \mathbb{Z}_{2^n}$. To analyse the mixed-strategy version of this subgame, we utilise the notion of payoff matrices in game theory. Essentially they are matrices M_C and M_{Adv} of size $n \times 2^n$ such that $u_C(i,j) = (M_C)_{i,j}$ and $u_{\mathsf{Adv}}(i,j) = (M_{\mathsf{Adv}})_{i,j}$, respectively. This also means that given $\boldsymbol{p}_C$ and $\boldsymbol{p}_{\mathsf{Adv}}$ be column vectors of probabilities representing the mixed strategies of the client and the adversary, respectively, then their mixed-strategy utilities are respectively

$$U_C(\boldsymbol{p}_C, \boldsymbol{p}_{\mathsf{Adv}}) = \boldsymbol{p}_C^{\mathrm{T}} M_C \boldsymbol{p}_{\mathsf{Adv}} \quad \text{and} \quad U_{\mathsf{Adv}}(\boldsymbol{p}_C, \boldsymbol{p}_{\mathsf{Adv}}) = \boldsymbol{p}_C^{\mathrm{T}} M_{\mathsf{Adv}} \boldsymbol{p}_{\mathsf{Adv}}.$$

We start the proof by forming the payoff matrices for the client C and the adversary Adv. Let $\mathsf{one}(j)$ be the number of occurrences of 1 in the binary form of j, and the components of payoff matrices M_C and M_{Adv} at row $i \in \mathbb{Z}_n$ and column $j \in \mathbb{Z}_{2^n}$ are

$$(M_C)_{i,j} = \begin{cases} L_C^{imp} - nc_C & \text{if } 2^i \wedge j = 0 \\ L_C^{mitm} - nc_C & \text{otherwise} \end{cases}$$

$$(M_{\mathsf{Adv}})_{i,j} = \begin{cases} G_{\mathsf{Adv}}^{imp} - \mathsf{one}(j) n_S c_S & \text{if } 2^i \wedge j = 0 \\ G_{\mathsf{Adv}}^{mitm} - \mathsf{one}(j) n_S c_S & \text{otherwise} \end{cases}$$

where $G_{\mathsf{Adv}}^{imp} = -(L_C^{imp} + L_S^{imp})$ and $G_{\mathsf{Adv}}^{mitm} = -(L_C^{mitm} + L_S^{mitm})$. Let $\boldsymbol{p}_C$ be the client's mixed strategy with a uniform distribution, and $\boldsymbol{p}_{\mathsf{Adv}}$ be the adversary's mixed strategy. Then

$$U_{\mathsf{Adv}}(\boldsymbol{p}_C, \boldsymbol{p}_{\mathsf{Adv}}) = \boldsymbol{p}_C^{\mathrm{T}} M_{\mathsf{Adv}} \boldsymbol{p}_{\mathsf{Adv}} = \frac{1}{n} \left(\sum_{i=0}^{n-1} (M_{\mathsf{Adv}})_i \right) \boldsymbol{p}_{\mathsf{Adv}} \tag{5.11}$$

where $(M_{\mathsf{Adv}})_i$ denotes the row vector with index $i \in \mathbb{Z}_n$ of M_{Adv}. We further notice that for each column $j \in \mathbb{Z}_{2^n}$ of M_{Adv}, there are exactly $\mathsf{one}(j)$ components with value $G_{\mathsf{Adv}}^{mitm} - \mathsf{one}(j) n_S c_S$, and the rest are $G_{\mathsf{Adv}}^{imp} - \mathsf{one}(j) n_S c_S$. Hence, the sum of n components of columns j are

$$\sum_{i=0}^{n-1} (M_{\mathsf{Adv}})_{i,j} = \mathsf{one}(j) G_{\mathsf{Adv}}^{mitm} + (n - \mathsf{one}(j)) G_{\mathsf{Adv}}^{imp} - \mathsf{one}(j) n_S c_S n$$

$$= \mathsf{one}(j)(G^{mitm}_{\mathsf{Adv}} - n_S c_S n) + (n - \mathsf{one}(j))G^{imp}_{\mathsf{Adv}}$$

Denote by $\boldsymbol{p}'_{\mathsf{Adv}}$ the adversary's mixed strategy with 2^n components of the form $(1, 0, \ldots, 0)$. In words, by selecting $\boldsymbol{p}'_{\mathsf{Adv}}$ the adversary's realised strategy is $d = 0b0\ldots0$, i.e., it never attempts an oMitM attack. It is easy to see that $U_{\mathsf{Adv}}(\boldsymbol{p}_C, \boldsymbol{p}'_{\mathsf{Adv}}) = G^{imp}_{\mathsf{Adv}}$. Consider $G^{mitm}_{\mathsf{Adv}} - G^{imp}_{\mathsf{Adv}} \leq n_S c_S n$ we then have

$$\begin{aligned}
U_{\mathsf{Adv}}(\boldsymbol{p}_C, \boldsymbol{p}_{\mathsf{Adv}}) &= \frac{1}{n}\left(\sum_{i=0}^{n-1}(M_{\mathsf{Adv}})_i\right)\boldsymbol{p}_{\mathsf{Adv}} \\
&= \frac{1}{n}\sum_{j=0}^{2^n-1}(\boldsymbol{p}_{\mathsf{Adv}})_j\left(\mathsf{one}(j)(G^{mitm}_{\mathsf{Adv}} - n_S c_S n) + (n - \mathsf{one}(j))G^{imp}_{\mathsf{Adv}}\right) \\
&\leq \frac{1}{n}\sum_{j=0}^{2^n-1}(\boldsymbol{p}_{\mathsf{Adv}})_j\left(\mathsf{one}(j)G^{imp}_{\mathsf{Adv}} + (n - \mathsf{one}(j))G^{imp}_{\mathsf{Adv}}\right) \\
&= G^{imp}_{\mathsf{Adv}} = U_{\mathsf{Adv}}(\boldsymbol{p}_C, \boldsymbol{p}'_{\mathsf{Adv}}).
\end{aligned}$$

Therefore $\boldsymbol{p}'_{\mathsf{Adv}}$ is always the adversary's best response. In addition, the client's mixed-strategy utility is always $U_C(\boldsymbol{p}_C, \boldsymbol{p}'_{\mathsf{Adv}}) = L^{imp}_C - nc_C$ for all $\boldsymbol{p}_C$. Therefore $(\boldsymbol{p}_C, \boldsymbol{p}'_{\mathsf{Adv}})$. Further, if $G^{mitm}_{\mathsf{Adv}} - G^{imp}_{\mathsf{Adv}} < n_S c_S n$ then $\boldsymbol{p}'_{\mathsf{Adv}}$ becomes the adversary's strictly best response.

Similarly, denote by $\boldsymbol{p}''_{\mathsf{Adv}}$ the adversary's mixed strategy with 2^n components of the form $(0, \ldots, 0, 1)$. In words, by selecting $\boldsymbol{p}'_{\mathsf{Adv}}$ the adversary's realised strategy is $d = 0b1\ldots1$, i.e., it attempts oMitM attacks in all querying rounds with certainty. It is easy to see that $U_{\mathsf{Adv}}(\boldsymbol{p}_C, \boldsymbol{p}'_{\mathsf{Adv}}) = G^{mitm}_{\mathsf{Adv}} - n_S n_C n$. Consider $G^{mitm}_{\mathsf{Adv}} - G^{imp}_{\mathsf{Adv}} \geq n_S c_S n$ we likewise have

$$\begin{aligned}
U_{\mathsf{Adv}}(\boldsymbol{p}_C, \boldsymbol{p}_{\mathsf{Adv}}) &= \frac{1}{n}\left(\sum_{i=0}^{n-1}(M_{\mathsf{Adv}})_i\right)\boldsymbol{p}_{\mathsf{Adv}} \\
&= \frac{1}{n}\sum_{j=0}^{2^n-1}(\boldsymbol{p}_{\mathsf{Adv}})_j\left(\mathsf{one}(j)(G^{mitm}_{\mathsf{Adv}} - n_S c_S n) + (n - \mathsf{one}(j))G^{imp}_{\mathsf{Adv}}\right) \\
&\leq \frac{1}{n}\sum_{j=0}^{2^n-1}(\boldsymbol{p}_{\mathsf{Adv}})_j\left(\mathsf{one}(j)(G^{mitm}_{\mathsf{Adv}} - n_S c_S n) + (n - \mathsf{one}(j))(G^{mitm}_{\mathsf{Adv}} - n_S c_S n)\right) \\
&= G^{mitm}_{\mathsf{Adv}} - n_S c_S n = U_{\mathsf{Adv}}(\boldsymbol{p}_C, \boldsymbol{p}''_{\mathsf{Adv}})
\end{aligned}$$

This eventually implies that $(\boldsymbol{p}_C, \boldsymbol{p}''_{\mathsf{Adv}})$ is an equilibrium, and that $\boldsymbol{p}''_{\mathsf{Adv}}$ is a strictly best response when $G^{mitm}_{\mathsf{Adv}} - G^{imp}_{\mathsf{Adv}} > n_S c_S n$. This concludes both claims of the proposition.

□

The equilibria suggested in Lemma 5.4 provide strategies for the client and the attacker during the querying phase that match the requirement for a PBE. Indeed, given a belief about (n_C, c_C), (n_S, c_S) and $(n_{\mathsf{Adv}}, c_{\mathsf{Adv}})$, the client's strategy taken from an equilibrium is apparently optimal against the attacker's attack strategy. Likewise, given a belief about (n_C, c_C), (n_S, c_S), $(n_{\mathsf{Adv}}, c_{\mathsf{Adv}})$ and the client's strategy from an equilibrium, the attacker's attack strategy from that same equilibrium is also optimal. Nevertheless the equilibria indicates that the adversary would receive

$$u_{\mathsf{Adv}} = \max\left(-(L_C^{imp} + L_S^{imp}), -(L_C^{mitm} + L_S^{mitm}) - n_S c_S n^*\right)$$

as the outcome of the whole game. By employing backward induction on the game tree (Figure 5.10), we can see that the adversary would choose $(n_{\mathsf{Adv}}, c_{\mathsf{Adv}})$ such that n^* is minimised. This requires setting $n_{\mathsf{Adv}} \leq n_C$, thus making $n^* = n_C$, and we may as well assume that $n_{\mathsf{Adv}} = 1$, since any value of $n_{\mathsf{Adv}} \leq n_C$ would yield the same utility. What remains is the choice of $c_{\mathsf{Adv}}(n_C, c_C, n_S, c_S)$, which does not affect the adversary's utility, and therefore we assume that $c_{\mathsf{Adv}} = c_{\min}$. In overall, the optimal utility of the adversary is:

$$u_{\mathsf{Adv}} = \max\left(-(L_C^{imp} + L_S^{imp}), -(L_C^{mitm} + L_S^{mitm}) - n_S c_S n_C\right) \tag{5.12}$$

Here we assume that if there is a tie, then the adversary would prefer strategy $d = j = 0b0\ldots0$ (impersonation) over strategy $d = j = 0b1\ldots1$ (oMitM). At this stage, the client's utility, for each of its choice of (n_C, c_C) would then be

$$\begin{aligned} u_C = &- n_C \max(c_C, c_{\min}) \\ &+ \begin{cases} L_C^{mitm} & \text{if } L_C^{imp} + L_S^{imp} - L_C^{mitm} - L_S^{mitm} > n_S c_S n^*, \text{ or,} \\ L_C^{imp} & \text{otherwise.} \end{cases} \end{aligned} \tag{5.13}$$

Working backward another step, we need to analyse (5.13) to see how the client should pick n_C and c_C. Let $(n_C^{(1)}, c_C^{(1)})$ be the client's choice should it want oMitM to occur, and $(n_C^{(2)}, c_C^{(2)})$ otherwise. (5.13) can be rewritten as

$$u_C = \max\left(-c_C^{(1)} n_C^{(1)} + L_C^{mitm}, -c_C^{(2)} n_C^{(2)} + L_C^{imp}\right) \tag{5.14}$$

$$\text{where } [n_{\max}] \ni n_C^{(1)} < \left\lceil \frac{L_C^{imp} + L_S^{imp} - L_C^{mitm} - L_S^{mitm}}{n_S c_S} \right\rceil \leq n_C^{(2)} \in [n_{\max}]$$

Following the above expression, it is best to minimise the round complexity parameters,

and hence it is in the client's interest that

$$n_C^{(1)} = 1,\ n_C^{(2)} = \left\lceil \frac{L_C^{imp} + L_S^{imp} - L_C^{mitm} - L_S^{mitm}}{n_S c_S} \right\rceil^{3},\ \text{and}\ c_C^{(1)} = c_C^{(2)} = c_{\min}.$$

This gives the following optimal utility for the client:

$$u_C = \max\left(L_C^{mitm} - c_{\min}, L_C^{imp} - c_{\min} n_C^{(2)}\right) \tag{5.15}$$

Here also, we assume that if there is a tie, then the client would prefer to invoke more rounds to stop an oMitM attack over experiencing it. This would complete the construction of a subgame perfect equilibrium. For the reader, we summarise our analysis above in a lemma:

Lemma 5.5. *Let (n_S, c_S) be given, then the game between the client and the adversary whose extensive form depicted in Figure 5.10 has a perfect Bayesian equilibrium in which the client's utility is as in* (5.15)*.*

We now have characterised an equilibrium point for the client and the adversary given each choice of the server. To complete the equilibrium finding process, it is necessary to check which of the server's choice is also its best response against the corresponding equilibrium. Nevertheless, we summarise the extensive form of the whole game as follows:

1. Server: The server takes its action by selecting (n_S, c_S).
2. Client: Without observing (n_S, c_S), the client selects its pair of (n_C, c_C).
3. Adversary: On observing (n_S, c_S) and (n_C, c_C), the adversary picks $(n_{\mathsf{Adv}}, c_{\mathsf{Adv}})$.
4. Client: On observing $(n_{\mathsf{Adv}}, c_{\mathsf{Adv}})$, the client picks $r \in [\max(n_C, n_{\mathsf{Adv}})]$ as the place to hide q.
5. Adversary: Without observing r, the adversary picks $d \in \{0,1\}^{[\max(n_C, n_{\mathsf{Adv}})]}$.

We fully characterise desirable perfect Bayesian equilibria in the following proposition:

Proposition 5.3. *Assume that there exist $n_C, n_S \in [n_{\max}]$ and $c_S \in [c_{\min}, c_{\max}]$ such that $n_C n_S c_S \geq L_C^{imp} + L_S^{imp} - L_C^{mitm} - L_S^{mitm}$ and $L_C^{imp} - L_C^{mitm} \geq c_{\min}(n_C - 1)$, then a refined unauthenticated communication game has a perfect Bayesian equilibrium in which an oMitM attack does not occur, i.e., Definition 5.4 is not satisfied.*

[3] In order to compute $n_C^{(2)}$ the client needs to know (n_S, c_S), which is not possible, as the game tree shows. However, under the Perfect Bayesian Equilibrium, the client is allowed to select $n_C^{(2)}$ based on its belief about (n_S, c_S), which should be available when such equilibrium point is clearly specified.

Proof. The equilibrium involves the server's strategy (n_S, c_S), the client's strategy $(n_C, c_C = c_{\min}, r)$ where r is as in Lemma 5.4, and the adversary's strategy $(n_{\mathsf{Adv}} = 1, c_{\mathsf{Adv}} = c_{\min}, d_1, \ldots, d_{n_{\max}})$ where d_i are as in Lemma 5.4. Following Lemma 5.5, the conditions $n_C n_S c_S \geq L_C^{imp} + L_S^{imp} - L_C^{mitm} - L_S^{mitm}$ and $L_C^{imp} - L_C^{mitm} \geq c_{\min}(n_C - 1)$ imply that if we assign (n_S, c_S) to be the server's strategy, then the rest of players' strategies form a perfect Bayesian equilibrium. On the other hand, the server's utility is L_S^{imp}, which is its maximum possible utility value, and thus (n_S, c_S) is thus part of the equilibrium. □

Corollary 5.1. *Assume that $n_{\max} = c_{\max} = +\infty$ and $c_{\min} = 0$, then a refined unauthenticated communication game has a perfect Bayesian equilibrium in which an oMitM attack does not occur.*

5.5.3 Solutions for Adversaries with Feedbacks

In this subsection we consider an extra situation which might arise in reality: the adversary is able to tell if it has received the actual query, based on its communication with the server. We call this a *feedback adversary*. It is best to motivate this via an example. Indeed, suppose that the server has to perform some extra processing on the query before returning the response. This extra processing, though, is independent of the response construction, as it only delays the response. The adversary knows the construction of the response, but does not necessarily know the extra processing, and thus may use the delay time as an advantage in guessing if the query it made to the server somehow relates to the client's actual query, as opposed to being some random garbage. To deal with this extra issue we need to introduce the attacker's advantage into the model.

We start with the game definition as in Definition 5.6 with further restrictions on the strategies of the client and the adversary as in Lemma 5.3 (Figure 5.10). Our modification affects the adversary's behaviour after round complexity parameters have been exchanged. When (n_C, c_C), (n_S, c_S) and $(n_{\mathsf{Adv}}, c_{\mathsf{Adv}})$ are given, instead of picking $d \in \{0,1\}^{n^*}$ at once, the value of each $d_i \in \{0,1\}$ is decided as below:

$$d_i := d_i \left(\langle b_{n_{\mathsf{Adv}}, c_{\mathsf{Adv}}}(j) d_j \rangle_{j \in [i-1]} \right) \quad \text{where} \quad b_{n_{\mathsf{Adv}}, c_{\mathsf{Adv}}}(j) = \begin{cases} 0 & \text{if } j = r(n_{\mathsf{Adv}}, c_{\mathsf{Adv}}), \text{ or,} \\ 1 & \text{otherwise.} \end{cases} \tag{5.16}$$

In other words, if the adversary consults the server at round j, then it receives a feedback bit $b_{n_{\mathsf{Adv}}, c_{\mathsf{Adv}}}(j)$ of whether the client's query at this round is q, i.e., whether

$j = r(n_{\mathsf{Adv}}, c_{\mathsf{Adv}})$. If the adversary does not mount an oMitM attack at round j, then $b_{n_{\mathsf{Adv}},c_{\mathsf{Adv}}}(j)d_j$ gives no information since $d_j = 0$. Otherwise it conveys the feedback via the value of $b_{n_{\mathsf{Adv}},c_{\mathsf{Adv}}}(j)$. Also, the adversary's decision in each round theoretically depends on feedbacks it received from all previous rounds. We now provide equilibria condition as in the previous analysis, starting with a modification of Lemma 5.4:

Lemma 5.6. *Let (n_C, c_C), (n_S, c_S) and $(n_{\mathsf{Adv}}, c_{\mathsf{Adv}})$ be given, so that $r \in [n_{\max}]$ and $(d_i)_{i\in[n^*]}$ are the remaining strategies to pick, where $n^* = \max(n_C, n_{\mathsf{Adv}})$. Let $G = L_C^{imp} + L_S^{imp} - L_C^{mitm} - L_S^{mitm}$ and $c = n_S c_S$. Let a_C be the client's mixed strategy such that $\Pr[r = k] = c'/(G+(n^*-k)c')$, where $c' > 0$ satisfies that $\sum_{j=1}^{n^*} c'/(G+(n^*-j)c') = 1$. Then,*

- *if and only if $\sum_{k=0}^{n^*-1} c/(G + k \cdot c) \geq 1$ there exists a Nash equilibrium of the form $(a_C, (d_i)_{i\in[n^*]})$ in which $d_i = 0$ for all $i \in [n^*]$ (no oMitM attack).*
- *if and only if $\sum_{k=0}^{n^*-1} c/(G + k \cdot c) \leq 1$ there exists a Nash equilibrium of the form $(a_C, (d_i)_{i\in[n^*]})$ in which $d_i \left(\langle b_{n_{\mathsf{Adv}},c_{\mathsf{Adv}}}(j)d_j\rangle_{j\in[i-1]}\right) = 1$ iff $b_{n_{\mathsf{Adv}},c_{\mathsf{Adv}}}(j)d_j = 0$ for all $j \in [i-1]$ (oMitM attack success with certainty).*

Proof. We first analyse the pure strategies of C and Adv. The strategy for C remains the same as before, since C does not receive any information during the game play. Adv's action seems to be more complicated, as it depends on history, i.e., the value r decided by the client's strategy i, as depicted in (5.16). We prove that this complication can be alleviated. For simplicity of presentation we let $n = n^* = \max(n_{\mathsf{Adv}}, n_C)$ and $c = n_S c_S$.

We show that the adversary's strategy can be conveniently represented by a number $d \in \mathbb{Z}_n$ much like in Lemma 5.4 when it has no feedback. This is equivalent to showing that at each of the n rounds, the adversary does not have to provide different decisions against different client's strategies $r \in [n]$. We show this by induction. For convenience we denote by *forward* the adversary's decision to mount an oMitM attack at a round, and by *respond* the act of impersonating the server S. For round 1, Adv receives the first query from C, and thus its decision is independent of the client's strategy r.

Suppose our claim holds for round k, consider round $k+1$. If Adv responds (without querying S) at round k, then Adv does not receive any feedback, and consequently no further information about C's strategy r, and thus its decision at round $k+1$ is independent of r. Otherwise, if Adv mounts an attack at round k, then it receives a feedback that either the real query q or a random query $\boldsymbol{q}_k$ was sent in round k. In the former case, Adv can end the game by responding (without querying S) to all the remaining queries, that is, to set $d_m = 0$ for all $m \in \{k+1, \ldots, n\}$, because

due to Definition 5.4 Adv already succeeded an oMitM attack, and thus any later communication to S is unnecessary. Since this is without doubt the optimal strategy, the adversary Adv needs not make a decision. In the latter case, Adv receives a new query $\boldsymbol{q}_{k+1}$, and has to decide whether to forward ($d_{k+1} = 1$) or respond ($d_{k+1} = 0$). This means that the adversary needs to plan its decision for only one case, i.e., when $r > k$, which proves our claim by the principle of mathematical induction.

Consequently, we can represent Adv's strategy by $d \in \mathbb{Z}_{2^n}$, where for each $k \in [n]$, the k-th bit d_k of d means the following (assuming always $d_{-1} = 0$):

- $d_k = 0$: if $d_{k-1} = 0$, then respond; if $d_{k-1} = 1$ and feedback is 0, then responds in all remaining rounds, i.e., $\{k+1, \ldots, n\}$; otherwise, respond in this round r (and move to the next round).
- $d_k = 1$: if $d_{k-1} = 0$, then forward; if $d_{k-1} = 1$ and feedback is 0, then responds in all remaining rounds, i.e., $\{k+1, \ldots, n\}$; otherwise, forward in this round r (and move to the next round).

For compatibility with Lemma 5.4 and its proof, we replace the adversary's strategy d by symbol $j \in \mathbb{Z}_{2^n}$, and the client's strategy r by $i = n - r$, so that $i \in \mathbb{Z}_n$. In this case i represents the number of remaining rounds after the real query q was sent by the client C. The difference of this case compared to Lemma 5.4 is in Adv' utility. Indeed, suppose for example that $n = 3$, C's strategy is $i = 1$ (sending q at round $r = 3 - 1 = 2$) and Adv's strategy is $j = 3 = 0b011$. In words, C sends q in the second round, whereas Adv responds in the first round, forwards in the second round, and forwards again in the third round if the second round query is not q, that is, if $2^i \wedge j = 0$. In Lemma 5.4 Adv's utility would be $G^{mitm}_{\mathsf{Adv}} - 2c$, whereas here it would be $G^{mitm}_{\mathsf{Adv}} - c$ because Adv will respond in the last round, knowing that the second round query is q (since $2^i \wedge j = 2^1 \wedge 3 = 2 \neq 0$). While M_C is identical to that in Lemma 5.4, M_{Adv} is such that

$$(M_{\mathsf{Adv}})_{i,j} = \begin{cases} G^{imp}_{\mathsf{Adv}} - \mathsf{one}(j)c & \text{if } 2^i \wedge j = 0 \\ G^{mitm}_{\mathsf{Adv}} - \mathsf{one}(j \gg i)c & \text{otherwise} \end{cases}$$

where $\gg$ is the right-bit-shift operator.

Let $c' > 0$ be such that $\sum_{k=0}^{n-1} c'/(G^{mitm}_{\mathsf{Adv}} - G^{imp}_{\mathsf{Adv}} + k \cdot c') = 1$. Let column vector $\boldsymbol{p}_C$ with n components be C's strategy, such that $(\boldsymbol{p}_C)_i = c'/(G^{mitm}_{\mathsf{Adv}} - G^{imp}_{\mathsf{Adv}} + i \cdot c')$ for $0 \leq i \leq n-1$. Clearly we have $\sum \boldsymbol{p}_i = 1$ and $\boldsymbol{p}_i > \boldsymbol{p}_j$ for $i < j$. In words, q appears more and more likely toward the end of n rounds. In response to this, the best strategy for Adv is to communicate with S at later rounds rather than early ones. In fact, it

is obvious that, among all strategies j, j' of Adv such that $\mathsf{one}(j) = \mathsf{one}(j')$, then Adv would prefer j over j' if $j \leq j'$, since

$$G^{mitm}_{\mathsf{Adv}} - \mathsf{one}(j \gg i)c \geq G^{mitm}_{\mathsf{Adv}} - \mathsf{one}(j' \gg i)c.$$

This means that the adversary would prefer j of the form $0b0 \ldots 01 \ldots 1$, for all j of the above form. Thus, to prove the lemma statement it is sufficient to prove that strategies of this form give Adv no better utility than simply responding to all queries $(d = 0b0 \ldots 0)$. Let $G_1 = G^{mitm}_{\mathsf{Adv}}$ and $G_0 = G^{imp}_{\mathsf{Adv}}$, the column vector of M_{Adv} for each such j is

$$(G_1 - \mathsf{one}(j)c, G_1 - (\mathsf{one}(j) - 1)c, \ldots, G_1 - c, G_0 - \mathsf{one}(j)c, \ldots, G_0 - \mathsf{one}(j)c)$$

Let $\boldsymbol{p}^{(j)}_{\mathsf{Adv}}$ denote a mixed strategy of the adversary that assigns j with probability 1. It is easy to see that $U_{\mathsf{Adv}}(\boldsymbol{p}_C, \boldsymbol{p}^{(0)}_{\mathsf{Adv}}) = G_0$. Consider $\sum_{k=0}^{n-1} c/(G^{mitm}_{\mathsf{Adv}} - G^{imp}_{\mathsf{Adv}} + k \cdot c) \geq 1$, which implies $c' \leq c$. Then the adversary's mixed-strategy utility is

$$\begin{aligned}
&U_{\mathsf{Adv}}(\boldsymbol{p}_C, \boldsymbol{p}^{(j)}_{\mathsf{Adv}}) \\
=\ &\boldsymbol{p}_C^{\mathrm{T}}\ M_{\mathsf{Adv}}\ \boldsymbol{p}^{(j)}_{\mathsf{Adv}} = \sum_{k=0}^{n-1} (\boldsymbol{p}_C)_k (M_{\mathsf{Adv}})_{j,k} \\
=\ &\sum_{k=0}^{\mathsf{one}(j)-1} \frac{c'(G_1 - c(\mathsf{one}(j) - k))}{G_1 - G_0 + c'k} + \sum_{k=\mathsf{one}(j)}^{n-1} \frac{c'(G_0 - c \cdot \mathsf{one}(j))}{G_1 - G_0 + c'k} \\
=\ &\sum_{k=0}^{\mathsf{one}(j)-1} \frac{c'(G_1 - c'(\mathsf{one}(j) - k))}{G_1 - G_0 + c'k} + \sum_{k=\mathsf{one}(j)}^{n-1} \frac{c'(G_0 - c' \cdot \mathsf{one}(j))}{G_1 - G_0 + c'k} \\
&- (c - c') \left(\sum_{k=0}^{\mathsf{one}(j)-1} \frac{c'(\mathsf{one}(j) - k)}{G_1 - G_0 + c'k} + \sum_{k=\mathsf{one}(j)}^{n-1} \frac{c' \cdot \mathsf{one}(j)}{G_1 - G_0 + c'k} \right) \\
\leq\ &\sum_{k=0}^{\mathsf{one}(j)-1} \frac{c'(G_1 - c'(\mathsf{one}(j) - k))}{G_1 - G_0 + c'k} + \sum_{k=\mathsf{one}(j)}^{n-1} \frac{c'(G_0 - c' \cdot \mathsf{one}(j))}{G_1 - G_0 + c'k} \\
=\ &\sum_{k=0}^{\mathsf{one}(j)-1} \frac{c'(G_1 - G_0 - c'(\mathsf{one}(j) - k))}{G_1 - G_0 + c'k} + \sum_{k=\mathsf{one}(j)}^{n-1} \frac{c'(-c'\mathsf{one}(j))}{G_1 - G_0 + c'k} + G_0 \\
=\ &\sum_{k=0}^{\mathsf{one}(j)-1} \frac{c'(G_1 - G_0 + c'k)}{G_1 - G_0 + c'k} + \sum_{k=0}^{n-1} \frac{c'(-c'\mathsf{one}(j))}{G_1 - G_0 + c'k} + G_0 \\
=\ &c'\mathsf{one}(j) - c'\mathsf{one}(j) + G_0 = G_0 = U_{\mathsf{Adv}}\left(\boldsymbol{p}_C, \boldsymbol{p}^{(0)}_{\mathsf{Adv}}\right)
\end{aligned}$$

This thus implies that $(\boldsymbol{p}_C, \boldsymbol{p}^{(0)}_{\mathsf{Adv}})$ is a Nash equilibrium since given $\boldsymbol{p}^{(0)}_{\mathsf{Adv}}$ the client's utility is independent of its strategy. Also, when $\sum_{k=0}^{n-1} c/(G^{mitm}_{\mathsf{Adv}} - G^{imp}_{\mathsf{Adv}} + k \cdot c) > 1$ we have $\boldsymbol{p}^{(0)}_{\mathsf{Adv}}$ as the adversary's strictly best response against $\boldsymbol{p}_C$.

Next we consider the other case, i.e., $\sum_{k=0}^{n-1} c/(G^{mitm}_{\mathsf{Adv}} - G^{imp}_{\mathsf{Adv}} + k \cdot c) \leq 1$, which implies $c \leq c'$. Inheriting the above manipulation we have

$$
\begin{aligned}
&U_{\mathsf{Adv}}(\boldsymbol{p}_C, \boldsymbol{p}^{(j)}_{\mathsf{Adv}}) \\
&= G_0 + (c' - c) \left(\sum_{k=0}^{\mathsf{one}(j)-1} \frac{c'(\mathsf{one}(j) - k)}{G_1 - G_0 + c'k} + \sum_{k=\mathsf{one}(j)}^{n-1} \frac{c' \cdot \mathsf{one}(j)}{G_1 - G_0 + c'k} \right) \\
&= G_0 + (c' - c) \left(\sum_{k=0}^{\mathsf{one}(j)-1} \frac{c'(n - k)}{G_1 - G_0 + c'k} + \sum_{k=\mathsf{one}(j)}^{n-1} \frac{c'(n - k)}{G_1 - G_0 + c'k} \right) \\
&\quad - (c' - c) \left(\sum_{k=0}^{\mathsf{one}(j)-1} \frac{c'(n - \mathsf{one}(j))}{G_1 - G_0 + c'k} + \sum_{k=\mathsf{one}(j)}^{n-1} \frac{c'(n - k - \mathsf{one}(j))}{G_1 - G_0 + c'k} \right) \\
&= U_{\mathsf{Adv}}\left(\boldsymbol{p}_C, \boldsymbol{p}^{(0b1\ldots1)}_{\mathsf{Adv}}\right) - (c' - c) \left(n - \mathsf{one}(j) - \sum_{k=\mathsf{one}(j)}^{n-1} \frac{c'k}{G_1 - G_0 + c'k} \right) \\
&\leq U_{\mathsf{Adv}}\left(\boldsymbol{p}_C, \boldsymbol{p}^{(0b1\ldots1)}_{\mathsf{Adv}}\right) - (c' - c)\left(n - \mathsf{one}(j) - (n - \mathsf{one}(j))\right) = U_{\mathsf{Adv}}\left(\boldsymbol{p}_C, \boldsymbol{p}^{(0b1\ldots1)}_{\mathsf{Adv}}\right)
\end{aligned}
$$

Therefore $\left(\boldsymbol{p}_C, \boldsymbol{p}^{(0b1\ldots1)}_{\mathsf{Adv}}\right)$ is a Nash equilibrium. Moreover, when $\sum_{k=0}^{n-1} c/(G^{mitm}_{\mathsf{Adv}} - G^{imp}_{\mathsf{Adv}} + k \cdot c) > 1$ we also have $\boldsymbol{p}^{(0b1\ldots1)}_{\mathsf{Adv}}$ as the adversary's strictly best response. Together with the previous conclusion we can infer the claims of the proposition. □

Given this result, we again perform backward induction in a similar manner to the previous subsection, by analysing how round parameters are chosen given that the client and the adversary behave as in Lemma 5.6. Moving a step upward the game tree Figure 5.10, the adversary would need to pick (n', c') knowing that its utility would be

$$
u_{\mathsf{Adv}} = \max\left(-(L^{imp}_C + L^{imp}_S), -(L^{mitm}_C + L^{mitm}_S) - \sum_{k=1}^{n^*} \frac{c \cdot k \cdot c'}{G + (n^* - k)c'} \right) \tag{5.17}
$$

It is thus best for the adversary to select $n' = 1$ so that $n^* = \max(n_C, n')$ is minimised, i.e., $n^* = n_C$. Again, for simplicity we also set $c' = c_{\min}$, as the choice of c' does not affect Adv's utility. Working backward the game tree another step, the above choice of

(n', c') yields the client's utility as

$$u_C = -\, n_C \max(c_C, c_{\min}) + \begin{cases} L_C^{mitm} & \text{if } \sum_{k=0}^{n^*-1} c/(G + k \cdot c) < 1 \text{ or,} \\ L_C^{imp} & \text{otherwise.} \end{cases}$$

where $n^* = \max(n', n_C) = n_C$. Let $(n_C^{(1)}, c_C^{(1)})$ and $(n_C^{(2)}, c_C^{(2)})$ be as before, then their optimal values would be

$$n_C^{(1)} = 1,\ n_C^{(2)} = \overline{n},\ \text{and } c_C^{(1)} = c_C^{(2)} = c_{\min}. \tag{5.18}$$

where $\overline{n} \in \mathbb{N}$ be minimum such that $\sum_{k=0}^{\overline{n}-1} c/(G + k \cdot C) \geq 1$. The client's utility can then be shortened as in (5.15), giving the following alternative version of Proposition 5.3:

Proposition 5.4. *Assume that there exist $n_C, n_S \in [n_{\max}]$ and $c_S \in [c_{\min}, c_{\max}]$ such that $\sum_{k=0}^{n_C} c/(G + k \cdot c) \geq 1$ for $c = n_S c_S$ and $G = L_C^{imp} + L_S^{imp} - L_C^{mitm} - L_S^{mitm}$ and $L_C^{imp} - L_C^{mitm} \geq c_{\min}(n_C - 1)$, then a refined unauthenticated communication game with feedback adversary has a perfect Bayesian equilibrium in which an oMitM attack does not occur, i.e., Definition 5.4 is not satisfied.*

Proof. The proof for this is similar to that for Proposition 5.3, where server's strategy is (n_S, c_S), the client's strategy is $(n_C, c_C = c_{\min}, r)$ where r is as in Lemma 5.5, and the adversary's strategy is $(n_{\mathsf{Adv}} = 1, c_{\mathsf{Adv}} = c_{\min}, d_1, \ldots, d_{n_{\max}})$ with d_i are as in Lemma 5.5. □

Corollary 5.2. *Assume that $n_{\max} = c_{\max} = +\infty$ and $c_{\min} = 0$, then a refined unauthenticated communication game with feedback adversary has a perfect Bayesian equilibrium in which an oMitM attack does not occur.*

5.6 Protocol Implementation

In our game model, we make use of several assumptions about the communication among protocol participants. Particularly, we assume with respect to Definition 5.2 that message transmission is atomic, that is, either no information or the whole message is conveyed. We also assume in Definition 5.4 that the communication relevant to an oMitM is revealed in full to the adversary, which could be considered too strict for an adversary model. Also, in the specification of the client and the server' strategies, we assume the unrealistic existence of store-then-forward bridges among the players, as

well as the enforcement of cost when sending a query. All of these assumptions are meant to simplify the model so that analysis is tractable.

In this section, we discuss techniques that would help realising above assumptions. The main idea is to use cryptographic tools in restricting players' abilities, so that they would eventually behave in ways stated in the assumptions. Since we use cryptographic techniques under computational security rather than information-theoretic security, our implementation would introduce some negligible advantages to game players. This would result in at most some negligible gain in utility once a player deviates from prescribed equilibria computed in Section 5.5. However, we do not discuss such situations as they can be trivially captured by replacing the traditional notion of equilibrium to computational one, e.g., ϵ-Nash equilibrium.

5.6.1 Definitions of Security

Our first step is to convert above assumptions into formal objectives. These objectives allow us to later verify that our cryptographic implementation of Qry (Figure 5.6) and Res (Figure 5.7) protocols reconciles with such assumptions. Informally the assumptions include:

- Atomic message transmission: a message destined to one party should be easily intercepted by another. The interceptor, or adversary, would then have no reason not to capture the whole message before performing any further action.

- Costly querying: the descriptions for Send_c and $\mathsf{Receive}_c$ respectively used in Figure 5.6 and Figure 5.7 requires that, even with knowledge from past queries and current help from the client, if the adversary does not incur a cost c, the server would not accept any query. However, we note that if the server rejects the query, the adversary receives no useful information, which is same as if it did not at all communicate with the server. Therefore, we may relax the above requirement: if the adversary's cost is $c \cdot p$ for some probability p, then the server would accept the query with probability at most p. Given that, the adversary's cheapest way to cheat is to randomise between honestly executing Send_c and not communicating with the server at all. Such randomisation is properly considered in our game analysis. In this section we thus focus on satisfying this relaxation. There are different types of cost as mentioned in Section 5.4.1. As an example, in this section we use proof of work in the form of computation cost.

- Store-then-forward bridges: because such bridges are unrealistic, an implementation of delaying message transmission must occur solely between the sender and

the receiver, and thus must ensure that during the delaying period, the intended recipient of the message must neither learn any information about the message (*hiding*) nor be able to produce any related message (*non-malleability*), and that the sender must not be able to modify the content of the message during the delaying period (*binding*).

We translate these descriptions into formal definitions as below:

Definition 5.7. *An atomic message transmission process is a tuple of PPT algorithms* $(\mathsf{Send}, \mathsf{Receive})$ *such that for all strings* m *and* dst *of polynomially-bounded lengths there exists a PPT receiver* Adv *with*

$$\Pr[m' \leftarrow \mathsf{Adv}^{\mathsf{Send}^{dst}(m)} : m = m'] = 1$$

Definition 5.8. *Denote by* $\mathsf{cost}(\mathsf{Alg})$ *a realisation of the computational cost when executing a probabilistic algorithm* Alg*. A secure costly message transmission process is a tuple of PPT algorithms* $(\mathsf{Send}, \mathsf{Receive})$ *satisfying for some negligible function* ϵ *that for all strings* m*,* src *and* dst *of polynomially-bounded lengths the following properties are satisfied:*

- *Correctness:*

$$\Pr\left[\begin{array}{c}(sig_1, m') \leftarrow (\mathsf{Send}_{c,n}^{dst}(m), \mathsf{Receive}_{c,n}^{src}) : \\ sig_1 = \mathsf{true} \wedge m' = m \wedge \mathsf{cost}(\mathsf{Send}_{c,n}^{dst}(m)) \geq c - \epsilon(n)\end{array}\right] = 1,$$

- *Message indistinguishability: if the adversary does not tamper with the communication messages, then it can infer no information about the high-level about the protocol message being sent, i.e., for all PPT adversaries* $\mathsf{Adv} = (\mathsf{Adv1}, \mathsf{Adv2})$ *and all cost values* $c \in [c_{\min}, c_{\max}]$*, define the following experiment:*

$$\Pi_{\mathsf{IND\text{-}MSG}}^{\mathsf{Adv}} = \left[\begin{array}{c}(m_0, m_1, s) \leftarrow \mathsf{Adv1}^{\mathsf{Send}_{n,\cdot}'(\cdot)}(n, c); b \leftarrow_{\$} \{0,1\}; \\ (\cdot,\cdot) \leftarrow (\mathsf{Send}_{n,c}^{dst}(m_b), \mathsf{Receive}_{n,c}^{src}); b' \leftarrow \mathsf{Adv2}(n, c, s, \mathsf{tr})\end{array}\right]$$

where tr *denotes the message transcripts of the communication produced by* $\mathsf{Send}_{n,c}^{dst}(m_b)$ *and* $\mathsf{Receive}_{n,c}^{src}$*, then*

$$\Pr\left[\Pi_{\mathsf{IND\text{-}MSG}}^{\mathsf{Adv}} : b = b'\right] \leq 1/2 + \epsilon(n), \tag{5.19}$$

- *Costly modification: if the adversary tampers with the communication, in order to either capture the high-level protocol message being sent, or to modify it, then it would cost the same as the adversary trying to send a message itself, i.e., for all PPT adversaries* $\mathsf{Adv} = (\mathsf{Adv1}, \mathsf{Adv2})$ *and all cost values* $c \in [c_{\min}, c_{\max}]$ *define the following experiment:*

$$\Pi^{\mathsf{Adv}}_{\mathsf{MOD\text{-}MSG}} = \left[\begin{array}{c} (s, m) \leftarrow \mathsf{Adv1}^{\mathsf{Send}^{\cdot}_{n,\cdot}(\cdot)}(n, c); (sig, \cdot, m') \leftarrow \\ \\ (\mathsf{Send}^{dst}_{n,c}(m), \mathsf{Adv2}(n, c, s), \mathsf{Receive}^{src}_{n,c}) \end{array}\right]$$

then for all probabilities $p \in [0, 1]$

$$\Pr\left[\Pi^{\mathsf{Adv}}_{\mathsf{MOD\text{-}MSG}} :\perp \neq m' \mid \mathsf{cost}(\mathsf{Adv2}) \leq c \cdot p \wedge \mathsf{tr}_S \neq \mathsf{tr}_R\right] \leq p + \epsilon(n) \tag{5.20}$$

where tr_S *and* tr_R *are message transcripts in the view of* $\mathsf{Send}^{dst}_{n,c}(m)$ *and* $\mathsf{Receive}^{src}_{n,c}$*, respectively.*

Definition 5.9. *Let* $\mathsf{Send} = (\mathsf{Send1}, \mathsf{Send2})$ *and* $\mathsf{Receive} = (\mathsf{Receive1}, \mathsf{Receive2})$ *be tuples of PPT algorithms*[4]*. Then* $(\mathsf{Send}, \mathsf{Receive})$ *is a secure delayed message transmission process if there exists a negligible function* ϵ *such that for all strings* m*,* src *and* dst *of polynomially-bounded lengths the following properties are satisfied:*

- *Correctness:*

$$\Pr\left[\begin{array}{c} (t, s) \leftarrow (\mathsf{Send1}^{dst}_n(m), \mathsf{Receive1}^{src}_n); (\cdot, m') \leftarrow \\ \\ (\mathsf{Send2}^{dst}_n(m, t), \mathsf{Receive2}^{src}_n(s)) : m = m' \end{array}\right] = 1$$

- *Hiding: for all stateful PPT adversaries* Adv

$$\Pr[(m_0, m_1) \leftarrow \mathsf{Adv}(dst); b \leftarrow_{\$} \{0, 1\}; b' \leftarrow \mathsf{Adv}^{\mathsf{Send1}^{dst}_n(m_b)} : b = b'] \leq 1/2 + \epsilon(n)$$

- *Binding: for all stateful PPT adversaries* Adv

$$\Pr\left[\begin{array}{c} s \leftarrow \mathsf{Adv}(dst);\ (m, \cdot) \leftarrow (\mathsf{Receive2}^{src}_n(s), \mathsf{Adv}); \\ \\ (m', \cdot) \leftarrow (\mathsf{Receive2}^{src}_n(s), \mathsf{Adv}) : m \neq m' \wedge m, m' \neq \perp \end{array}\right] \leq \epsilon(n)$$

[4]Here Send1 and Receive1 resemble the transmission of a message from the sender to the bridge, and Send2 and Receive2 represent forwarding it by the bridge to the receiver.

- *Non-malleability: for every stateful PPT adversary* Adv*, there exists a PPT adversary* Adv' *such that for all valid efficiently sampleable distribution* $\mathcal{D}$ *and all polynomial-time computable relation* R

$$\Pr\left[\Pi^{\mathsf{Adv},n,R,\mathcal{D}}_{\mathsf{NM\text{-}Delay}} : R(m,m')=1\right] - \Pr\left[\Pi^{\mathsf{Adv}',n,R,\mathcal{D}}_{\mathsf{NM\text{-}Delay\text{-}Sim}}\right] \leq \epsilon(n),$$

where

$$\Pi^{\mathsf{Adv},n,R,\mathcal{D}}_{\mathsf{NM\text{-}Delay}} = \left[\begin{array}{c} m \leftarrow \mathcal{D}; (s_1,\cdot,t_1) \leftarrow (\mathsf{Receive1}^{src}_n, \mathsf{Adv}(src,dst), \mathsf{Send1}^{dst}_n(m)); \\ (m',\cdot,\cdot) \leftarrow (\mathsf{Receive2}^{src}_n(s_1), \mathsf{Adv}(src,dst), \mathsf{Send2}^{dst}_n(m,t_1)) \end{array}\right]$$

$$\Pi^{\mathsf{Adv}',n}_{\mathsf{NM\text{-}Delay\text{-}Sim}} = \left[m \leftarrow \mathcal{D}; m' \leftarrow \mathsf{Adv}'(src,dst,n) : R(m,m')=1\right]$$

5.6.2 Protocol Construction

Our final step is to construct the process of sending and receiving messages in Figure 5.6 (client's strategies)and Figure 5.7 (server's strategies) using cryptographic techniques mentioned in Section 1.3. The simplest way of sending messages that satisfies Definition 5.7 is to transmit them in plaintext. That way the adversary can effortlessly intercept any message destined to any party. The problem is that most likely it would not support Definition 5.8 and Definition 5.9 which require some form of binding between a message being sent and the actual sender and receiver. Indeed, whilst Definition 5.8 requires that the cost of querying cannot be transferred between client-adversary and adversary-server communications, Definition 5.9 also desires that a delayed message from the server to the adversary must not be passable to the client.

The above observation suggests a need to securely bind the knowledge of the sender and the receiver to the communication between them. We achieve this using a secure key-exchange protocol (Definition 1.16) and an authenticated encryption mechanism (Definition 1.18). The former facilitates the binding by forming a key between the sender and the receiver, and the latter ensures that an adversary must properly execute key exchange in order to receive the incoming message. All communication would then be encrypted and authenticated. The details of the protocol are provided in Figure 5.11, where "$\mathsf{Enc}_k(\cdot)$:" (resp. "$\mathsf{Dec}_k(\cdot)$:") indicates a step in which a message is encrypted then sent (resp. received then decrypted) using key k agreed previously. Likewise,"$\mathsf{Enc}_k(\cdot), \mathsf{Dec}_k(\cdot)$:" indicates a step in which a number of encrypted messages are sent and received. The security of this protocol is provided below.

Proposition 5.5. *Assume that* (I,R) *is a secure-key exchange protocol,* $(\mathsf{Prove},\mathsf{Verify})$ *is a secure proof-of-work mechanism, and* $(\mathcal{K},\mathsf{Enc},\mathsf{Dec})$ *with* $\mathcal{K}$*'s output generated by*

Parameters: (I, R), $(\mathsf{Prove}, \mathsf{Verify})$, $(\mathcal{K}, \mathsf{Enc}, \mathsf{Dec})$.

Protocol.

$\mathsf{Send}^{dst}_{n,c}(m)$:
- $k \leftarrow \mathsf{I}(n)$
- **if** k is not valid **then return** $\perp$
- $\mathsf{Enc}_k(\cdot), \mathsf{Dec}_k(\cdot)$: $sig \leftarrow \mathsf{Prove}(c, k)$
- **if** $sig \neq \mathsf{true}$ **then return** $\perp$
- $\mathsf{Enc}_k(\cdot)$: m
- **return** true

$\mathsf{Receive}^{src}_{n,c}$:
- $k \leftarrow \mathsf{R}(n)$
- **if** k is not valid **then return** $\perp$
- $\mathsf{Enc}_k(\cdot), \mathsf{Dec}_k(\cdot)$: $sig \leftarrow \mathsf{Verify}(c, k)$
- **if** $sig \neq \mathsf{true}$ **then return** $\perp$
- $\mathsf{Dec}_k(\cdot)$: m
- **return** m

Figure 5.11: Protocol Send and Receive for messages other than responses r_i.

(I, R) *is a secure authenticated encryption mechanism with respect to some message space* $\mathcal{M}$*, then the tuple* $(\mathsf{Send}, \mathsf{Receive})$ *as in Figure 5.11 is an atomic (Definition 5.7) and secure costly (Definition 5.8) message transmission process.*

Proof. Before proceeding with the proof, we provide the following supporting lemma:

Lemma 5.7. *Assume that* (I, R) *is a key-exchange protocol that satisfies the correctness and key indistinguishability of Definition 1.16 and* $(\mathcal{K}, \mathsf{Enc}, \mathsf{Dec})$ *with* $\mathcal{K}$*'s output generated by* (I, R) *is a secure authenticated encryption mechanism with respect to some message space* $\mathcal{M}$*, for all stateful PPT adversaries* Adv *define the following experiment*

$$\Pi^{\mathsf{Adv}}_{\mathsf{MOD\text{-}SUP}} = \left[(k, \cdot, k') \leftarrow (\mathsf{I}(n), \mathsf{Adv}, \mathsf{R}(n)); c \leftarrow \mathsf{Adv}^{\mathsf{Enc}_k(\cdot)}\right]$$

there exists a negligible function ϵ *such that*

$$\Pr\left[\Pi^{\mathsf{Adv}}_{\mathsf{MOD\text{-}SUP}} : \mathsf{Dec}_k(c) \neq \perp \wedge\ c \notin C \wedge k = k'\right] \leq \epsilon(n)$$

where C *is the set of ciphertexts output by* $\mathsf{Enc}_k(\cdot)$*.*

Proof. We first assume that $\Pr\left[\Pi^{\mathsf{Adv}}_{\mathsf{MOD\text{-}SUP}} : k = k'\right] = \epsilon_1(n)$ is non-negligible. Given $k = k'$, we notice that $\Pi^{\mathsf{Adv}}_{\mathsf{MOD\text{-}SUP}}$ is the same as the IND-CTXT experiment in Definition 1.18, except that k is generated by $(\mathsf{I}(n), \mathsf{Adv}, \mathsf{R}(n))$ characterised by some distribution

$\mathcal{D}'$ (given $k = k'$) instead of $(\mathsf{I}(n), \mathsf{R}(n))$ with distribution $\mathcal{D}$. We also have

$$\begin{aligned}
&\Pr[\Pi^{\mathsf{Adv}}_{\text{MOD-SUP}} : \mathsf{Dec}_k(c) \neq \perp \wedge c \notin C \mid k = k'] \qquad (5.21)\\
&= \sum_{k \in \mathcal{D}'} \Pr[c \leftarrow \mathsf{Adv}^{\mathsf{Enc}_k(\cdot)} : \mathsf{Dec}_k(c) \neq \perp \wedge c \notin C] = \epsilon_2(n),\\
&\Pr[\text{IND-CTXT succeeds}] \qquad (5.22)\\
&= \sum_{k \in \mathcal{D}} \Pr[c \leftarrow \mathsf{Adv}^{\mathsf{Enc}_k(\cdot)} : \mathsf{Dec}_k(c) \neq \perp \wedge c \notin C] \leq \epsilon(n)
\end{aligned}$$

Note that $\Pr[\text{IND-CTXT succeeds}] \leq \epsilon(n)$ due to the fact that $(\mathcal{K}, \mathsf{Enc}, \mathsf{Dec})$ is secure given that $\mathcal{K}$'s output is generated by (I, R). Suppose that ϵ_2 is non-negligible, then $\Pr\left[\Pi^{\mathsf{Adv}}_{\text{MOD-SUP}} : \mathsf{Dec}_k(c) \neq \perp \wedge c \notin C \mid k = k'\right] - \Pr\left[\text{IND-CTXT succeeds}\right]$ is non-negligible. In other words, there exists some key k'' such that $\Pr_{\mathcal{D}'}[k = k''] - \Pr_{\mathcal{D}}[k = k''] > 0$ is non-negligible. However, this means that we can use Adv as a valid adversary against the key-exchange experiment in Definition 1.16, who returns 0 whenever given $k_b = k''$. The attack success of such adversary is:

$$\begin{aligned}
&\Pr\left[\begin{array}{l} (k_0, \cdot, k_2) \leftarrow (\mathsf{I}(n), \mathsf{Adv}, \mathsf{R}(n)); k_1 \leftarrow_{\mathcal{D}} \mathcal{K}; \\ b \leftarrow_{\$} \{0,1\}; b' \leftarrow \mathsf{Adv}(k_b) : b = b' \mid k_0 = k_2 \end{array}\right]\\
&\geq 1/2 + 1/2\ \epsilon_1(n)(\Pr_{\mathcal{D}'}[k = k'] - \Pr_{\mathcal{D}}[k = k'])
\end{aligned}$$

which violates the assumption that (I, R) is a secure key-exchange protocol since $\Pr[k_0 = k_2] = \epsilon_1(n)$ which is non-negligible. Hence ϵ_2 must be negligible, and so is the following:

$$\Pr\left[\Pi^{\mathsf{Adv}}_{\text{MOD-SUP}} : \mathsf{Dec}_k(c) \neq \perp \wedge c \notin C \wedge k = k'\right] = \epsilon_1(n)\epsilon_2(n)$$

□

The proof for *atomicity* is trivial since the behaviour of Send and $\mathsf{Receive}$ does not depend on *src* and *dst*, therefore the adversary Adv can effortlessly execute $\mathsf{Receive}$ and capture m. The *correctness* property is also straightforward. Indeed, because $\mathsf{I}(n)$ and $\mathsf{R}(n)$ are executed correctly, the two ends receive the same key k. Because $(\mathsf{Prove}, \mathsf{Verify})$ is a secure proof-of-work mechanism, and that $\mathsf{Prove}(c, k)$ and $\mathsf{Verify}(c, k)$ are executed correctly, they both produce $sig_1 = \mathsf{true}$ deterministically, and that the cost of executing $\mathsf{Prove}(c, k)$ is bounded below by $c - \epsilon(n)$. Finally, the test for $sig = \mathsf{true}$ in $\mathsf{Send}^{dst}_{n,c}(m)$ is passed, and thus m is sent over, ensuring that the receiver gets $m' = m$ correctly. This concludes the proof of correctness.

Due to the correctness of the key exchange protocol (I, R), both ends would receive

the same key k. The correctness of secure querying in $(\mathsf{Prove}, \mathsf{Verify})$ guarantees that $\mathsf{cost}(\mathsf{Send}^{dst}_{n,c}(m)) \geq \mathsf{cost}(\mathsf{Prove}(c,k)) \geq c - \epsilon(n)$. It also indicates that both $\mathsf{Prove}(c,k)$ and $\mathsf{Verify}(c,k)$ returns true, leading to $\mathsf{Send}^{dst}_{n,c}(m)$ and $\mathsf{Receive}^{src}_{n,c}$ returning true and $m' = m$, respectively. Thus the correctness property is complete. Also, the proof for *message indistinguishability* comes straightforwardly from the IND-CPA property of encryption, and thus can be omitted.

To prove the *costly modification* property, we consider several different modifications the adversary can make to the communication between $\mathsf{Send}^{dst}_{n,c}(m)$ and $\mathsf{Receive}^{src}_{n,c}$. For simplicity of presentation we always assume $\mathsf{cost}(\mathsf{Adv2}) \leq c \cdot p \wedge \mathsf{tr}_S \neq \mathsf{tr}_R$ and omit it in all probability expressions. Let k_S and k_R be the session key perceived by $\mathsf{Send}^{dst}_{n,c}(m)$ and $\mathsf{Receive}^{src}_{n,c}$, respectively. Let $\boldsymbol{k}_S$ be the set of keys perceived by all executions of Send in the experiment, with $k_S \in \boldsymbol{k}_S$. Consider the following cases:

- $k_R \notin \boldsymbol{k}_S$: we note from the construction of $\mathsf{Receive}$ in Figure 5.11 that $m' \neq \perp$ in the experiment $\Pi^{\mathsf{Adv}}_{\mathsf{MOD\text{-}MSG}}$ implies that the proof-of-work verifier in $\mathsf{Receive}$ returns $sig_1 = \mathsf{true}$. We also note that because $k_R \notin \boldsymbol{k}_S$ the adversary does not interact with any Prove oracle with input $id = k_R$. Therefore, due to the verifiability property of proof-of-work (Definition 1.17) there exists a negligible function ϵ_1 such that

$$\begin{aligned}
&\Pr\left[\Pi^{\mathsf{Adv}}_{\mathsf{MOD\text{-}MSG}} : m \neq \perp \mid k_R \notin \boldsymbol{k}_S\right] \\
&\leq \Pr\left[\Pi^{\mathsf{Adv}}_{\mathsf{MOD\text{-}MSG}} : sig_1 = \mathsf{true} \mid k_R \notin \boldsymbol{k}_S\right] \\
&\leq p + \epsilon_1(n)
\end{aligned}$$

- $k_R \in \boldsymbol{k}_S$: we break this down further to two sub-cases:
  - $k_S = k_R \in \boldsymbol{S}$: assume that this occurs with non-negligible probability, then due to the synchronisation property of key exchange (Definition 1.16) the adversary must not have modified the key-exchange communication. Therefore, in order for $\mathsf{tr}_S \neq \mathsf{tr}_R$ to hold, Adv must modify the consequent encrypted messages. Lemma 5.7 however guarantees that $\mathsf{Receive}$ would accept any modified ciphertext with negligible probability, and hence there exists a negligible function ϵ_2 such that

$$\Pr\left[\Pi^{\mathsf{Adv}}_{\mathsf{MOD\text{-}MSG}} : m \neq \perp \mid k_S = k_R \in \boldsymbol{S}\right] \Pr[k_S = k_R \in \boldsymbol{S}] = \epsilon_2(n)$$

  - $k_S \neq k_R \in \boldsymbol{S}$: to be able to ensure that $m' \neq \perp$, the adversary must convince the Verify routine in $\mathsf{Receive}$ to return $sig_2 = \mathsf{true}$. We also note that because

$k_R \neq k_S$ the adversary does not interact with any Prove oracle with input $id = k_R$ while communicating with Prove. Therefore,

$$\begin{aligned}
&\Pr\left[\Pi^{\mathsf{Adv}}_{\text{MOD-MSG}} : m \neq \perp \mid k_S \neq k_R \in \boldsymbol{S}\right] \\
&\leq \Pr\left[\Pi^{\mathsf{Adv}}_{\text{MOD-MSG}} : sig_2 = \mathsf{true} \mid k_S \neq k_R \in \boldsymbol{S}\right] \\
&\leq p + \epsilon_1(n)
\end{aligned}$$

In overall we thus have:

$$\begin{aligned}
\Pr\left[\Pi^{\mathsf{Adv}}_{\text{MOD-MSG}} : m \neq \perp\right] =& \Pr\left[\Pi^{\mathsf{Adv}}_{\text{MOD-MSG}} : m \neq \perp \mid k_R \notin \boldsymbol{k}_S\right] \Pr[k_R \notin \boldsymbol{k}_S] \\
&+ \Pr\left[\Pi^{\mathsf{Adv}}_{\text{MOD-MSG}} : m \neq \perp \mid k_S = k_R \in \boldsymbol{S}\right] \Pr[k_S = k_R \in \boldsymbol{S}] \\
&+ \Pr\left[\Pi^{\mathsf{Adv}}_{\text{MOD-MSG}} : m \neq \perp \mid k_S \neq k_R \in \boldsymbol{S}\right] \Pr[k_S \neq k_R \in \boldsymbol{S}] \\
\leq& \; p + \epsilon_3(n)
\end{aligned}$$

for some negligible function ϵ_3. □

We notice that in our design of Send and Receive, we have to perform key exchange for every message transmission. This is in fact unnecessary, as the client and the server can reuse the same key for every message transmission within the same protocol execution. Our proof of security above also supports this as it implies stricter security by mean of less flexibility to the adversary. We thus assume that this is the case from now on. Let k_C and k_S be the keys perceived by the client and the server, respectively. We moreover assume that $k_C \neq k_S$ since Lemma 5.7 shows that if $k_C = k_S$ then no adversary is able to obtain any information. In fact, the best it can do is to either behave as a router or simply drop the whole communication.

The remaining part of this protocol to construct is the message delayed transmission procedures, i.e., (Send1, Send2) and (Receive1, Receive2) that implements the bridges in Figure 5.8. The detailed algorithms for these procedures are given in Figure 5.12. The security of this construction heavily relies on the properties of non-malleable commitments, which can be shown below.

Proposition 5.6. *Assume that* (Setup, Commit, Open) *is a non-malleable commitment scheme. Let* $\mathsf{CK} \leftarrow \mathsf{Setup}(n)$ *be a common reference string, and* $k_S, k_C \neq \perp$ *be parameters embedded in* Send *and* Receive *such that* $k_S \neq k_C$ *in the presence of an adversary and* $k_S = k_C$ *otherwise. Then the tuple* (Send, Receive) *as in Figure 5.12 is an atomic (Definition 5.7) and secure delayed (Definition 5.9) message transmission process.*

Parameters: $n > 0$, $(\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open})$, $\mathsf{CK} \leftarrow \mathsf{Setup}(n)$.

Protocol.

$\mathsf{Send}^{dst}_{n,k_S}(m)$:
- $\mathsf{Send1}^{dst}_{n,k_S}(m)$:
  - **if** $k_S = \perp$ **then return** $\perp$
  - $(c, d) \leftarrow \mathsf{Commit}_{\mathsf{CK}}(k_S || m)$
  - $\mathsf{Enc}_{k_S}(\cdot) : c$
  - $t \leftarrow d$
  - **return** t
- $\mathsf{Send2}^{dst}_{n,k_S}(m, t)$:
  - $d \leftarrow t$
  - **if** $k_S = \perp$ **then return** $\perp$
  - $\mathsf{Enc}_{k_S}(\cdot) : d$
  - **return** true

$\mathsf{Receive}^{src}_{n,k_C}$:
- $\mathsf{Receive1}^{src}_{n,k_C}$:
  - **if** $k_C = \perp$ **then return** $\perp$
  - $\mathsf{Dec}_{k_C}(\cdot) : c$
  - $s \leftarrow c$
  - **return** s
- $\mathsf{Receive2}^{dst}_{n,k_C}(s)$:
  - $c \leftarrow s$
  - **if** $k_C = \perp$ **then return** $\perp$
  - $\mathsf{Dec}_{k_C}(\cdot) : d$
  - $k' || m \leftarrow \mathsf{Open}_{\mathsf{CK}}(c, d)$
  - **if** $k' \neq k_C$ **return** $\perp$
  - **return** m

Figure 5.12: Protocol Send and Receive for delaying responses r_i.

Proof. As usual, the correctness property is rather straightforward to prove. Note that in the correctness experiment there is no adversary, and therefore $k_S = k_C = k$. In this case it relies on the correctness of commitment and encryption/decryption. Further, the fact that *dst* and *src* are not used anywhere in the protocol description, therefore the atomicity property automatically applies.

In proving the hiding property, let $\Pi^{\mathsf{Adv}}_{\mathsf{Hide\text{-}Delay}}$ denote the hiding experiment under adversary Adv, and $\Pi^{\mathsf{Adv}'}_{\mathsf{Hide\text{-}Commit}}$ denote the hiding experiment in the commitment scheme under adversary Adv'. We assume that Adv succeeds in guessing with probability $1/2+\epsilon'(n)$ for some non-negligible function ϵ'. Then we can construct an adversary Adv' against $\Pi^{\mathsf{Adv}'}_{\mathsf{Hide\text{-}Commit}}$ from an adversary Adv against $\Pi^{\mathsf{Adv}}_{\mathsf{Hide\text{-}Delay}}$ as follows:

1. Let Adv picks m_0, m_1.
2. Output $m'_0 = k_S||m_0$ and $m'_1 = k_S||m_1$.
3. Receive $c \leftarrow \mathsf{Commit}_{\mathsf{CK}}(m'_b)$.
4. Give $\mathsf{Enc}_{k_S}(c)$ to Adv and receives bit b'.
5. Return b'.

We notice the view of Adv in this case is exactly the same as in $\Pi^{\mathsf{Adv}}_{\mathsf{Hide\text{-}Delay}}$, meaning that it returns $b = b'$ with exactly the same probability, and therefore

$$\Pr\left[\Pi^{\mathsf{Adv}}_{\mathsf{Hide\text{-}Delay}} \text{ succeeds}\right] = \Pr\left[\Pi^{\mathsf{Adv}'}_{\mathsf{Hide\text{-}Commit}} \text{ succeeds}\right] = 1/2 + \epsilon'(n)$$

which contradicts with the hiding property of commitment. Therefore, the hiding property of this protocol holds.

The proof for the binding property is in a similar manner. Let $\Pi^{\mathsf{Adv}}_{\mathsf{Bind\text{-}Delay}}$ and $\Pi^{\mathsf{Adv}'}_{\mathsf{Bind\text{-}Commit}}$ denote the binding experiments for the delayed message transmission and commitment scheme, respectively. The three steps in the experiment $\Pi^{\mathsf{Adv}}_{\mathsf{Bind\text{-}Delay}}$ involves Adv giving out (c, d, d'). Assume that Adv succeeds with non-negligible probability $\epsilon'(n)$, then we construct Adv' against $\Pi^{\mathsf{Adv}'}_{\mathsf{Bind\text{-}Commit}}$ as follows:

1. Let Adv picks c, d, d' and gives c, $\mathsf{Enc}_{k_C}(d)$ and $\mathsf{Enc}_{k_C}(d')$ to Adv'.
2. Perform decryption to get c, d, d' and output them to $\Pi^{\mathsf{Adv}'}_{\mathsf{Commit\text{-}Bind}}$.

The success probability Adv' is then

$$\begin{aligned}&\Pr\left[\Pi^{\mathsf{Adv}'}_{\mathsf{Commit\text{-}Bind}} \text{ succeeds}\right]\\ &= \Pr\left[t \leftarrow \mathsf{Open}_{\mathsf{CK}}(c, d), t' \leftarrow \mathsf{Open}_{\mathsf{CK}}(c, d') : t \neq t' \wedge t, t' \neq \perp\right]\end{aligned}$$

$$\geq \Pr\left[k||m \leftarrow t, k'||m' \leftarrow t' : t \neq t' \wedge t, t' \neq \perp \wedge m \neq m' \wedge k = k' = k_C\right]$$
$$= \Pr\left[m \neq m' \wedge k = k' = k_C\right]$$
$$= Pr\left[\Pi^{\mathsf{Adv}}_{\mathsf{Delay\text{-}Bind}} \text{ succeeds}\right] = \epsilon'(n).$$

This again contradicts with the binding property of commitment, and therefore the binding property of the delayed message transmission protocol must also hold. The last part of the proof is the non-malleability property. Assume that Adv has non-negligible advantage in producing related messages, i.e., for all simulator Sim

$$\Pr\left[\Pi^{\mathsf{Adv},n,R,\mathcal{D}}_{\mathsf{NM\text{-}Delay}} : R(m, m') = 1\right] - \Pr\left[\Pi^{\mathsf{Adv}',n,R,\mathcal{D}}_{\mathsf{NM\text{-}Delay\text{-}Sim}}\right] \geq \epsilon'(n) \tag{5.23}$$

for some non-negligible function ϵ'. For convenience we recall the operation of Adv (following Definition 5.12) below:

1. Receives (from $\mathsf{Send1}$) an encrypted commit value c_1.
2. Output (to $\mathsf{Receive1}$) an encrypted commit value c_2.
3. Receives (from $\mathsf{Send2}$) an encrypted decommit value d_1.
4. Output (to $\mathsf{Receive2}$) an encrypted commit value d_2.

Define Adv_1 as a slight modification of Adv as follows: whenever Adv outputs $c_2 = c_1$, then Adv_1 outputs $c_2 \neq c_1$ and later d_2 such that $(c_2, d_2) \leftarrow \mathsf{Commit}_{CK}(k'||m')$ such that $k' \neq k_C$. Due to the binding property, if Adv outputs $c_2 = c_1$ then it would lead to $\mathsf{Receive2}$ receiving $k_S||m'$, and would eventually output $\perp$ because the check $k_S = k_C$ fails. In the same vein Adv_1 would also lead to $\mathsf{Receive2}$ outputting $\perp$. Thus Adv_1 also satisfies (5.23). Consider experiment $\Pi^{\mathsf{Adv}',n,R',\mathcal{D}'}_{\mathsf{NM\text{-}Commit}}$ with the adversary Adv' constructed as follows:

1. Define R' as below:

$$R'(k||m, k'||m') = \begin{cases} 1 & \text{if } k_S = k \wedge k_C = k' \wedge R(m, m') = 1, \text{ or} \\ 1 & \text{if } k_S = k \wedge k_C \neq k' \wedge R(m, \perp) = 1, \text{ or} \\ 0 & \text{otherwise.} \end{cases}$$

2. Define distribution $\mathcal{D}'$ of $k||m$ such that $\Pr_{\mathcal{D}'}[k_S||m] = \Pr_{\mathcal{D}}[m]$ for all messages m sampleable by $\mathcal{D}$.
3. Sample $\mathcal{D}'$ for $m = k||\overline{m}$ (which guarantees $k = k_S$), produce $(c_1, d_1) \leftarrow \mathsf{Commit}_{\mathsf{CK}}(k||\overline{m})$, send c_1 to Adv_1, who would give $c_2 \neq c_1$ back, then output c_2.

4. Send d_1 to Adv_1 and receives back d_2 from Adv_1, then output d_2.

5. Compute $k'||\overline{m}' = m' \leftarrow \mathsf{Open}_{\mathsf{CK}}(c_2, d_2)$.

The success probability of Adv' is analysed as follows:

$$\begin{aligned}
&\Pr\left[\Pi^{\mathsf{Adv}',n,R',\mathcal{D}'}_{\text{NM-Commit}} : c_1 \neq c_2 \wedge R'(m,m') = 1\right] \\
=&\Pr\left[k' = k_C \wedge R(\overline{m},\overline{m}') = 1\right] + \Pr\left[k' \neq k_C \wedge R(\overline{m},\perp) = 1\right] \\
=&\Pr\left[\Pi^{\mathsf{Adv}_1,n,R,\mathcal{D}}_{\text{NM-Delay}} : R(m_t,m'_t) = 1 \wedge m'_t \neq \perp\right] + \Pr\left[\Pi^{\mathsf{Adv}_1,n,R,\mathcal{D}}_{\text{NM-Delay}} : R(m_t,m'_t) = 1 \wedge m'_t = \perp\right] \\
=&\Pr\left[\Pi^{\mathsf{Adv}_1,n,R,\mathcal{D}}_{\text{NM-Delay}} : R(m_t,m'_t) = 1\right].
\end{aligned}$$

The above manipulation is made possible because the probability distribution over the pair (m_t, m'_t) is the same as that of the pair $(\overline{m}, \overline{m}')$ in experiment $\Pi^{\mathsf{Adv}_1,n,R,\mathcal{D}}_{\text{NM-Delay}}$, thank to the design of $\mathcal{D}'$. Then for every simulator Sim on the simulation experiment $\Pi^{\mathsf{Sim},n,R,\mathcal{D}}_{\text{NM-Delay-Sim}}$ in the delayed message transmission scheme define a simulator Sim' on the simulation experiment $\Pi^{\mathsf{Sim}',n,R',\mathcal{D}'}_{\text{NM-Commit-Sim}}$ in the commitment scheme, as follows:

- if Sim outputs $\overline{m}' \neq \perp$, then Sim' outputs $m' = k_C||\overline{m}'$,
- if Sim outputs $\overline{m}' = \perp$, then Sim' outputs $m' = k'||\overline{m}'$, with $k' \neq k_C$.

We then have:

$$\begin{aligned}
&\Pr\left[\Pi^{\mathsf{Sim}',n,R',\mathcal{D}'}_{\text{NM-Commit-Sim}} : R'(m,m') = 1\right] \\
=&\Pr\left[k' = k_C \wedge R(\overline{m},\overline{m}') = 1\right] + \Pr\left[k' \neq k_C \wedge R(\overline{m},\perp) = 1\right] \\
=&\Pr\left[\Pi^{\mathsf{Sim},n,R,\mathcal{D}}_{\text{NM-Delay-Sim}} : R(m_t,m'_t) = 1 \wedge m'_t \neq \perp\right] + \Pr\left[\Pi^{\mathsf{Sim},n,R,\mathcal{D}}_{\text{NM-Delay-Sim}} : R(m_t,m'_t) = 1 \wedge m'_t = \perp\right] \\
=&\Pr\left[\Pi^{\mathsf{Sim},n,R,\mathcal{D}}_{\text{NM-Delay-Sim}} : R(m_t,m'_t) = 1\right]
\end{aligned}$$

Again, this is made possible since the value of m_t is distributed according to $\mathcal{D}$, and the distribution over m'_t is the same as that output by Sim. This thus implies that

$$\begin{aligned}
&\Pr\left[\Pi^{\mathsf{Adv}',n,R',\mathcal{D}'}_{\text{NM-Commit}} : c_1 \neq c_2 \wedge R'(m,m') = 1\right] - \Pr\left[\Pi^{\mathsf{Sim}',n,R',\mathcal{D}'}_{\text{NM-Commit-Sim}} : R'(m,m') = 1\right] \\
=&\Pr\left[\Pi^{\mathsf{Adv}_1,n,R,\mathcal{D}}_{\text{NM-Delay}} : R(m_t,m'_t) = 1\right] - \Pr\left[\Pi^{\mathsf{Sim},n,R,\mathcal{D}}_{\text{NM-Delay-Sim}} : R(m_t,m'_t) = 1\right] \geq \epsilon'(n)
\end{aligned}$$

for every simulator Sim' and corresponding Sim, which is contradictory, and hence the non-malleability of the delayed message transmission must hold. □

5.7 Practical Considerations

In this section we discuss several concerns regarding the practicality of our protocol implementation as well as of the overall game solution. These involve situations that are not covered by definitions of both the game and the security, as well as limitation rooted from environment parameters.

5.7.1 Multiple Executions

In our game model we only provide security for the client C to make a query once. Security does not automatically expand when C makes the second query, or re-send the previous query should the first execution of Qry fails for some reason. This is because information exposed during the first execution may give Adv some advantage in guessing the value of subsequent queries.

Several examples of this problem exist. In one case, suppose the adversary knows that C would make the same query q every time. He would then perform oMitM attack for the first query in order to learn the value of q, even if it results in an expected loss for that communication session. However, in subsequent executions the adversary knows in which round the real query would occur, and only attack at that round. Thus Adv's utility now becomes more promising.

In general, security (by definition) for multiple executions is guaranteed as long as query indistinguishability is achieved, as in Proposition 5.2. Fortunately in most communication nowadays it is not easy to predict what would happen next in a conversation unless the adversary spends enough efforts to study the previous "discussions" as well as the parties involved. Our model is meant for unauthenticated communication, which mean that the purpose of communication is not very sensitive, e.g., WWW surfing. Therefore above "information gathering" efforts account for cost that might be unbearable to the adversary, and thus demotivate him.

Another technical problem is information leakage which might occur even if q is not learned. Consider an adversary Adv who aborts the protocol with C after it receives the first query $\boldsymbol{q}_1$. It costs Adv nothing to do so. However, there is $1/n^*$ chance that $\boldsymbol{q}_1$ is q. As the protocol is aborted, C may re-execute it until she succeeds. By carrying out this attack over and over again, Adv collects a set $\{\boldsymbol{q}_1^{(i)}\}$, where each $\boldsymbol{q}_1^{(i)}$ has $1/n^*$ chance of being q. Thus, with no expense Adv can be certain that $q \in \{\boldsymbol{q}_1^{(i)}\}$ with high probability. It may then ask S these queries to learn q.

The problem of leakage can be solved by persistently forcing the protocol to finish. In other words, if C experiences an abort, then in the next time C would continue from the last stage rather than restarting Qry. For security reason one needs to make sure

that an adversary cannot deny a protocol abort to C, nor can it claim a protocol abort to S while it did not happen. To do so, it is possible to have both parties signing the protocol execution as they proceed, using their own private key and a digital signature scheme. These signatures are exchanged, so that later each party has a proof from the other about the state of an execution.

5.7.2 Small Query/Response Spaces

The size of the query space Q may also influence the security of multiple executions. Indeed, if $\mathcal{Q}$ is small, then C's queries between two different executions may likely to coincide. Although not considered as a valid oMitM attack by our definition, in case the nature of the response is static, then if Adv encounters a query that was seen previously, Adv does not need to query S for a reply and thus saves some efforts. Our protocol is thus practically unsuitable for multiple executions when $\mathcal{Q}$ is small and responses are static, i.e., they do not change over time. Otherwise, an example of dynamic responses where our solution is applicable is: queries asking the result of a live football match.

While the query set $\mathcal{Q}$ is large in many conversations, e.g., human chat, remote control, WWW queries, the answers to queries might be restrictive. We exclude situations when there is only one answer, because they are rather one-way communication than query-response. In the worst case, a response may be either "yes" or "no". Thus Adv may simulate S's response with high probability. What the client C can do, however, is to execute Qry many times, each with a different query. This would lessen Adv's chance of giving all correct answers. In practice C can save these multiple executions by examining all answers in a single execution, as opposed to accepting only the answer to q as Qry does. This achieves the same effect with less cost.

5.7.3 Uninteresting Impersonations

In many scenarios, impersonation attacks are not attractive to adversaries. These include, for example, WWW surfing, private chat between people unknown to the adversary, remote login (apart from password stealing purposes). The main reason is that it is hard to simulate the other end's behaviour without causing suspicion. When this happen we may consider Adv payoff $G^{imp}_{\mathsf{Adv}} = -(L^{imp}_C + L^{imp}_S)$ negative, because he gains nothing while risking detection.

This means that if impersonation (with certainty) is the only available attack, Adv would rather choose to not attack at all. To capture this, we need to modify Adv's payoff matrix M_{Adv} (as in the proofs of Lemma 5.4 and Lemma 5.5) so that all values in the first column (originally G^{imp}_{Adv}) are set by 0 to reflect Adv's inactivity. Similarly, the first

column (originally $L_C^{imp} - nc_C$) of M_C must also be set by $-\max(n_C, n_S)\max(c_C, c_S)$. Assume that $-L_C^{mitm} \geq \max(n_C, n_S)\max(c_C, c_S) - n_C c_C$, that is, the benefit of being attack-free is greater than the extra cost for carrying out an attack-free communication, it is not difficult to prove that with these changes Lemma 5.4 and Lemma 5.5 still hold if G is replaced by G_{Adv}^{mitm} instead of the original value $G_{\mathsf{Adv}}^{mitm} - G_{\mathsf{Adv}}^{imp}$. Thus, given appropriate choices (which we discuss later) by C and S that validate the above assumption, no attack would happen. This also motivates our protocol construction.

5.7.4 Proof-of-Works May Fail

Several works have criticised the effectiveness of POW mechanisms in deterring attackers, notably [28, 88]. The main problem is the difference in *production frontiers*, which refers to the fact that the adversary and the user have different valuation of the same cost, i.e., cost metric. For example, while the user cannot afford 10 minutes of computing because he is in a hurry, the adversary can if he has more than one computer and can divide the work load, thus shortening the computing time. Even if parallel computing is not possible, the adversary can still survive 10 minutes computing if he has spare time to wait. As a result, the cost for POW to sufficiently deter attacks would surpass the client's payoff.

Our solution has a feature that addresses this issue. We first let $R_C \geq 1$ denote the ratio between C and Adv's valuations of cost. In other words, if a computing cost has value c to C, then it has value c/R_C to Adv. Proposition 5.3 and Proposition 5.4 give NEs in which Adv does no better with an oMitM attack. This is possible as $\mathsf{cost}_{\mathsf{Adv}}^{mitm} \geq G$, where $\mathsf{cost}_{\mathsf{Adv}}^{mitm}$ is the attack cost that guarantees an oMitM attack, and $G = L_C^{imp} + L_S^{imp} - L_C^{mitm} - L_S^{mitm}$. When the difference of production frontiers is taken into account, the new condition must be $\mathsf{cost}_{\mathsf{Adv}}^{mitm} \geq G \cdot R_C$. This means that the client C and the server S must change their choices of (n_C, c_C) and (n_S, c_S) to reflect this update, which most likely results in an increase for C's executing cost, which is at most $\mathsf{cost}_C = \max(n_C, n_S)\max(c_C, c_S)$ (occurs when there is no attack).

Moreover, our solution is only useful if $\min(0, L_C^{imp}) - L_C^{mitm} \geq \mathsf{cost}_C$. Let $R_U = G/(L_C^{imp} - L_C^{mitm}) \geq 1$, the following condition is necessary

$$\mathsf{cost}_{\mathsf{Adv}}^{mitm}/\mathsf{cost}_C \geq G \cdot R_C/(L_C^{imp} - L_C^{mitm}) = R_C R_U \tag{5.24}$$

and also this ratio should be as large as possible. There are at least two ways to raise attack cost (and hence executing cost), either via increasing the number of rounds $n^* = \max(n_C, n_{\mathsf{Adv}})$, or by introducing higher querying cost. However, only the former would help raising the ratio $\mathsf{cost}_{\mathsf{Adv}}^{mitm}/\mathsf{cost}_C$. Figure 5.13 shows how choices of n^* in

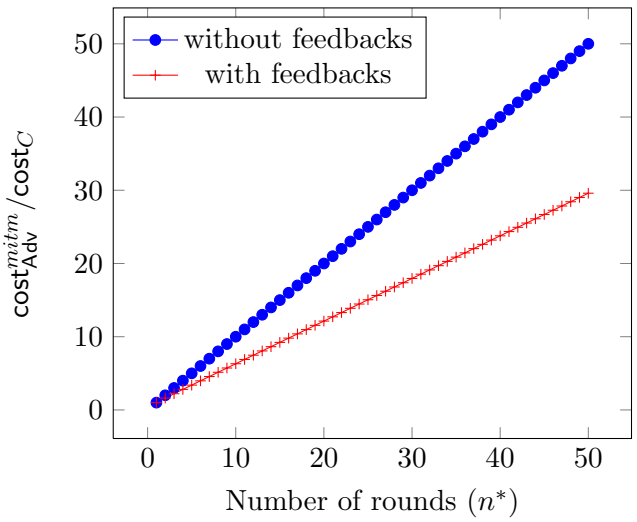


Figure 5.13: Adversary-user cost ratio

Proposition 5.3 and Proposition 5.4 affect this ratio. It is thus evident that with appropriate choices of n^* and c^* the aforementioned problem with POW can be overcome. Note that the cost ratio for adversary with feedbacks assumption is not linear in n^*, only approximately. Also, although theoretically one can raise n^* arbitrarily high to ensure (5.24), there are factors against that. First, raising n^* more than necessary would lead to prohibitive cost to the client C. Secondly, n^* should be bounded by the number of queries that can be made, i.e., $|\mathcal{Q}|$, or else repetitions of queries might be noticeable.

5.7.5 Bootstrapping of Security

The main practical limitation of our solution is that it is expensive due to the cost of querying. It is therefore necessary to mitigate its use once security has been confirmed. This is known as bootstrapping of security. A real-world example of bootstrapping is Secure Shells (SSH). In SSH, the security bootstrapping happens when a client connects to the server for the first time and receives its public key. The client must make a leap of faith that the key belongs to the server, instead of an adversary who luckily appears in the middle of the communication. This leap of faith is made once only, and future communication is authenticated based on the trusted key.

In this scenario, the client C may execute Qry once with as many rounds as possible to deter oMitM attacks. Suppose then, that either the adversary decide not to impersonate, or it does so but is then discovered by C (perhaps because the imposture is not plausible enough). In the latter case C is alerted and may find a more secure

route to communicate with the server S. In both cases C succeeds in bootstrapping security with S by, e.g., learning S's public key, as in the case of SSH. In subsequent communication, both parties would not need to use our solution any more.

There are a few problems with security bootstrapping, however. First, the adversary Adv may impersonate C and force S to bootstrap security. This allows Adv to query S in the future using normal methods that involve no cost. After that Adv can easily arrange an oMitM attack no matter how C executes Qry, since it needs not pay to query S. Secondly, even if Adv may not impersonate C to S, Adv may sacrifice by launching an oMitM attack during C's first communication with S, and suffer even enormously negative utility. However, after (in)security is bootstrapped, both C and S switch to normal communication, and Adv starts reaping positive utility as he stays persistently in between.

Our argument is that in many cases the same adversary may not stay persistently between C and S for longer than a period of, say length T. This length value may be lessened, e.g., when C and/or S are mobile. In that case, both C and S may set the bootstrapping period to be of length T, within which all communication must use our solution. This clearly eliminates Adv's advantage after security bootstrapping.

5.7.6 Attack Detection

The fact that our solution makes oMitM attacks more difficult than legitimate communication means that it might as well be used for MitM attack detection in addition to deterrence. This can be achieved, for example through recognition of excessive number of queries to S within a short time period, as they would be many times more than normal (Figure 5.13). The attacks may also be realised by C, e.g., it takes too long to receive a response, as Adv needs to execute Qry to communicate with the server S.

These uses of timing analysis to detect attacks may be defeated, for example if Adv is able to assume a different identity for every connection to S, or if it has fast enough computing power to shorten the execution time Qry, for example when querying requires proof-of-work in terms of computation. Otherwise, the success of detection would likely to reduce the cost of security bootstrapping, as attacks may be discovered before (Qry, Res) finishes being executed.

5.7.7 Examples of Application

The structure of $\mathcal{Q}$ should depend strongly on the public knowledge of the client C's potential query q before it is sent out. As a result, it is mostly bound to each individual server S, while in private conversation it may also be bound to each particular client

C. We illustrate this with two examples of application where our solution might be applied: search engine and password authentication.

Search engine. When S is the search engine, q is in the form of a search phrase. Depending on the search areas covered by S, the set $\mathcal{Q}$ should contain all possible phrases in those areas. On the other hand, if C's interest is well-known, e.g., C is a nurse, then $\mathcal{Q}$ should discard all phrases that relate to for instance computer science. When $\mathcal{Q}$ is very large, C does not need to store the whole $\mathcal{Q}$, but a subset of it is sufficient. An example could be a dictionary of words in nursing area, so that a random search phrase could be formed by several of these words together.

Password authentication. In password authentication, a query is essentially a password. Therefore $\mathcal{Q}$ should contain all plausible passwords that following a particular rule. Again, $\mathcal{Q}$ should be refined if information about C is exposed, e.g., C is English, then $\mathcal{Q}$ should not contain passwords with Norwegian words. In general case, $\mathcal{Q}$ needs not be stored, as passwords can be easily generated at random.

5.8 Conclusions

In this chapter we explore a topic that has not been well studied, i.e., security of unauthenticated two-party client-server communication. With strong definitions of adversaries and security used in cryptography or formal method security, it is generally hard to derive a satisfying solution. Alternatively, we define a weaker, but meaningful security notion, called online man-in-the-middle (oMitM) attacks, in which the attacker communicate with both ends in parallel, and effectively use information from each side in communicating with the other. For attacks that do not fall in this category we are regarded as impersonation.

By assuming a fixed payoff for the adversary in each type of attacks, as well as communicating parties' losses and utilities, we use game theory to model a game that captures interactions and behaviours of these entities. The game analysis points out solution concepts in the form of Nash equilibria that discourage man-in-the-middle attack. We implement the game by designing a communication protocol that equivalently captures the players' activities, with supports from cryptographic primitives. Finally, we discuss additional practicality features regarding our solution.

Our solution relies on the use of cost in querying and multi-round requests to make attacks more expensive than the communicating cost (Figure 5.13). It places the very first steps in dealing with the known presence of an attacker in unauthenticated communication. The communicating cost might be further optimised via bootstrapping of security and attack detection. Therefore, our solution is reasonable given that the

risk is certain. That said, it might even be tailored less costly for scenarios where the presence of an attacker is less certain (e.g. SSH remote login). This creates a motivation for future research in economic approaches to unauthenticated communication.

Chapter 6

Conclusion

Following the proliferation of information technologies in business routines, security of information infrastructures has become a major concern for societal entities, ranging from individuals to large organisations, as well as governmental and critical infrastructures. While traditional treatments of security have been well-established in both results and research community, the economics of information security bring a new awareness to the problem of securing information. Indeed, this approach addresses the rationality of participants involved in a security scenario, something which have been assumed otherwise by traditional security research. Although in its infant stage, this area of research possess many potentials, as the nature of security issues involves scenarios consisting of multiple independent and selfish decision makers. This is thus a "fertile" land for research that applies game-theoretic techniques to study potential behaviours of the relevant decision makers. This thesis is no more than an effort to contribute in developing further this new approach, with works on problems in different subfields of information security. Our research employs game-theoretic analysis to produce insights, as well as rational solutions for these problems.

Particularly, our work starts with addressing a high-level problem, i.e., organisational strategic security investments in protection against advanced persistent threats (APTs). We extend FlipIt, a game of timing model developed by Dijk et al. [141], by enriching it with different types of strategies. We then analyse, compare and contrast these strategies to provide insights on how organisations should make their strategic security plans in dealing with APTs. Moving to a more specific problem, in Chapter 3 we consider the needs for investing in security research, as well as sharing of security information among firms, especially in competitive environments. Here firms must consider the trade-offs between selfishly protecting themselves to earn competitiveness, or "open their hearts" for mutual improvements and better social welfare. Our analysis

leads to insights about how firms would behave under different conditions of competition, markets, and risks from security breaches. Next, in Chapter 4 we look at the problem of security for outsourced computation, i.e., the honesty of the contractors who carry out the computation. As solutions, we design principal-agent contracts that attract contractors, while at the same time encouraging them to be honest, as well as optimising the outsourcer's expenditure. We also consider the problem of information leakage and collusion among agents. Finally, Chapter 5 is dedicated to an investigation on a low-level problem: security of network communication. Here we study an issue seldomly considered in the research community: unauthenticated communication. As contributions, we formalise a relaxed yet meaningful notion of security, which otherwise cannot be satisfied when there is no authentication. From game-theoretic modelling and analysis, we develop a cryptographic protocol that rationally discourages the adversary from violating our prescribed security.

In summary, by studying a breadth of problems in information security, it is in our hope that the thesis contributes to demonstrating the potential of this research approach, especially on using game-theoretic analysis. Indeed, our works alone open a number of questions for future research, as can be seen throughout each chapter individually.

Bibliography

[1] M. Abliz and T. Znati. A guided tour puzzle for denial of service prevention. In *ACSAC '09*, pages 279–288, 2009. 149

[2] T. Alpcan and T. Başar. An intrusion detection game with limited observations. In *12th Int. Symp. on Dynamic Games and Applications*, Sophia Antipolis, France, July 2006. 26

[3] T. Alpcan and T. Başar. *Network Security: A Decision and Game Theoretic Approach*. Cambridge University Press, 2011. 26

[4] E. Altman, K. Avrachenkov, and A. Garnaev. Jamming in wireless networks: The case of several jammers. In *Game Theory for Networks, 2009. GameNets '09. International Conference on*, pages 585 –592, may 2009. 26

[5] P. Anand. Foundations of rational choice under risk. *OUP Catalogue*, 1995. 15

[6] R. Anderson and T. Moore. The economics of information security: A survey and open questions. In *Fourth bi-annual Conference on the Economics of the Software and Internet Industries*, Jan 2007. 14, 15

[7] M. J. Atallah, Y. Cho, and A. Kundu. Efficient data authentication in an environment of untrusted third-party distributors. In *IEEE ICDE*, 2008. 88

[8] S. Ba, J. Stallaert, A. B. Whinston, and H. Zhang. Choice of transaction channels: The effects of product characteristics on market evolution. *Journal of management information systems*, 21(4):173–197, 2005. 63

[9] R. Bace and P. Mell. *Intrusion detection systems*. NIST special publication. U.S. Dept. of Commerce, Technology Administration, National Institute of Standards and Technology, 2001. 26

[10] B. Barak, R. Canetti, Y. Lindell, R. Pass, and T. Rabin. Secure computation without authentication. *Journal of Cryptology*, 24(4):720–760, 2011. 31, 137, 140

[11] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear programming: theory and algorithms*. John Wiley & Sons, 2013. 97, 115

[12] R. Bejtlich. Testimony before the USCC Hearing on "Developments in China's Cyber and Nuclear Capabilities", March 26, 2012. `http://www.uscc.gov/hearings/2012hearings/written_testimonies/hr12_03_26.php`. 33

[13] S. Bekker. Gates: Microsoft's upping security R&D budget. `http://redmondmag.com/articles/2004/01/28/gates-microsofts-upping-security-rd-budget.aspx`. 66

[14] M. Belenkiy, M. Chase, C. C. Erway, J. Jannotti, A. Küpçü, and A. Lysyanskaya. Incentivizing outsourced computation. In *NetEcon*. ACM, 2008. 28, 88, 89

[15] M. Belenkiy, M. Chase, C. C. Erway, J. Jannotti, A. Küpçü, and A. Lysyanskaya. Incentivizing outsourced computation. In *NetEcon*. ACM, 2008. 90, 93, 95, 104, 123

[16] M. Bellare, J. Kilian, and P. Rogaway. The security of the cipher block chaining message authentication code. *Journal of Computer and System Sciences*, 61(3):362–399, 2000. 139

[17] M. Bellare and P. Rogaway. Entity authentication and key distribution. In *CRYPTO '93*, volume 773 of *LNCS*, pages 232–249. Springer-Verlag, 1994. 13

[18] M. Bellare and P. Rogaway. Entity authentication and key distribution. In *CRYPTO '93*, volume 773 of *LNCS*, pages 232–249. Springer-Verlag, 1994. 138

[19] D. Bellhouse. The problem of Waldegrave. *Journal Électronique d'Histoire des Probabilités et de la Statistique*, 3(2):null, 2007. 15

[20] C. G. Billo. Cyber warfare: An analysis of the means and motivations of selected nation states. Technical report, Institute for Security Technology Studies at Darthmouth College, 2004. 28, 33

[21] D. Blackwell. The noisy duel, one bullet each, arbitrary non-monotone accuracy. 1949. 28, 34

[22] R. Böhme. Security audits revisited. In A. Keromytis, editor, *Financial Cryptography and Data Security*, volume 7397 of *Lecture Notes in Computer Science*, pages 129–147. Springer Berlin Heidelberg, 2012. 27

[23] R. Böhme and M. Félegyházi. Optimal information security investment with penetration testing. In *GameSec*, pages 21–37, 2010. 34, 49, 54

[24] R. Böhme and T. Moore. The iterated weakest link: A model of adaptive security investment. In *Workshop on the Economics of Information Security (WEIS)*, 2009. 34, 54

[25] D. Boneh. The decision diffie-hellman problem. In *Algorithmic number theory*, pages 48–63. Springer, 1998. 216

[26] R. Branzei, D. Dimitrov, and S. Tijs. *Models in cooperative game theory*, volume 556. Springer Science & Business Media, 2008. 89

[27] S. Buchegger and T. Alpcan. Security games for vehicular networks. In *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, pages 244 –251, sept. 2008. 26

[28] L. J. Camp and D. Liu. Proof of work can work. In *Workshop on the Economics of Information Security (WEIS)*. TPRC, 2006. 149, 191

[29] K. Campbell, L. A. Gordon, M. P. Loeb, and L. Zhou. The economic cost of publicly announced information security breaches: empirical evidence from the stock market. *Journal of Computer Security*, 11(3):431–448, 2003. 61

[30] R. Canetti and H. Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. EUROCRYPT 2001, pages 453–474, London, UK, UK, 2001. Springer-Verlag. 23

[31] R. Canetti, B. Riva, and G. N. Rothblum. Practical delegation of computation using multiple servers. In *ACM CCS*, 2011. 88

[32] B. Carbunar and M. V. Tripunitara. Payments for outsourced computations. *IEEE Transactions on Parallel and Distributed Systems*, 23(2), 2012. 90

[33] M. C. Carlos, J. E. Martina, G. Price, and R. F. Custodio. A proposed framework for analysing security ceremonies. In P. Samarati, W. Lou, and J. Zhou, editors, *Proceedings of the 7th International Conference on Security and Cryptography*, SECRYPT 12, pages 440–445. SciTePress, jul 2012. 14

[34] H. Cavusoglu, B. Mishra, and S. Raghunathan. The effect of internet security breach announcements on market value: Capital market reactions for breached firms and internet security developers. *International Journal of Electronic Commerce*, 9(1):70–104, 2004. 61

[35] E. Chabrow. Identifying undetected breaches identifying undetected breaches: How data scientists analyze big data to spot vulnerabilities, April 20, 2012. http://www.bankinfosecurity.co.uk/interviews/identifying-undetected-breaches-i-1542. 33

[36] Y. E. Chan and K. E. Greenaway. Theoretical explanations for firms' information privacy behaviors. *Journal of the Association for Information Systems*, 6(6):7, 2005. 63

[37] H. Chen, X. Ma, W. Hsu, N. Li, and Q. Wang. Access control friendly query verification for outsourced data publishing. In *ESORICS*, 2008. 88

[38] E. Christoforou, A. F. Anta, C. Georgiou, M. A. Mosteiro, and A. Sánchez. Applying the dynamics of evolution to achieve reliability in master–worker computing. *Concurrency and Computation: Practice and Experience*, 2013. 90

[39] R. Cooper and T. W. Ross. Product warranties and double moral hazard. *The RAND Journal of Economics*, pages 103–113, 1985. 89

[40] A. Coviello. Open letter to RSA customers, March 17, 2011. http://www.rsa.com/node.aspx?id=3872. 28, 33, 35

[41] R. Croson, T. Boles, and J. K. Murnighan. Cheap talk in bargaining experiments: lying and threats in ultimatum games. *Journal of Economic Behavior & Organization*, 51(2):143–159, 2003. 89

[42] J. S. Demski and D. E. Sappington. Resolving double moral hazard problems with buyout agreements. *The RAND Journal of Economics*, pages 232–240, 1991. 89

[43] G. Di Crescenzo, Y. Ishai, and R. Ostrovsky. Non-interactive and non-malleable commitment. STOC '98, pages 141–150, New York, NY, USA, 1998. 25

[44] T. Dierks and E. Rescorla. RFC5246: The transport layer security (TLS) protocol version 1.2, aug 2008. 137

[45] D. Dolev, C. Dwork, and M. Naor. Nonmalleable cryptography. *SIAM review*, 45(4):727–784, 2003. 140

[46] D. Dolev, S. Even, and R. M. Karp. On the security of ping-pong protocols. *Information and Control*, 55(1):57–68, 1982. 139

[47] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29:198–208, 1983. 13, 139

[48] C. Dwork, A. Goldberg, and M. Naor. On memory-bound functions for fighting spam. In *CRYPTO 2003*, volume 2729 of *LNCS*, pages 426–444. Springer Berlin Heidelberg, 2003. 149

[49] C. Dwork and M. Naor. Pricing via processing or combatting junk mail. CRYPTO '92, pages 139–147, London, UK, UK, 1993. Springer-Verlag. 149

[50] C. Evans. Announcing project zero. `http://googleprojectzero.blogspot.com/2014/07/announcing-project-zero.html`. 29, 64

[51] J. Farrell. Meaning and credibility in cheap-talk games. *Games and Economic Behavior*, 5(4):514–531, 1993. 89

[52] E. Gal-Or. Information sharing in oligopoly. *Econometrica: Journal of the Econometric Society*, pages 329–343, 1985. 29, 63

[53] E. Gal-Or and A. Ghose. The economic incentives for sharing security information. *Information Systems Research*, 16(2):186–208, 2005. 29, 63

[54] X. Gao, W. Zhong, and S. Mei. Security investment and information sharing under an alternative security breach probability function. *Information Systems Frontiers*, pages 1–16. 27, 64

[55] R. Gennaro, C. Gentry, and B. Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In *CRYPTO*. Springer, 2010. 28, 88

[56] R. Gennaro, C. Gentry, and B. Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In *CRYPTO*. Springer, 2010. 30, 86, 90

[57] G. Gigerenzer and R. Selten. *Bounded rationality: The adaptive toolbox*. Mit Press, 2002. 15

[58] H. Gintis. *Game Theory Evolving: A Problem-Centered Introduction to Modeling Strategic Interaction*. Princeton University Press, 2009. 28, 90, 96, 100

[59] I. L. Glicksberg. A further generalization of the kakutani fixed point theorem, with application to nash equilibrium points. *Proceedings of the American Mathematical Society*, 3(1):170–174, 1952. 18

[60] S. Goel and H. A. Shawky. Estimating the market impact of security breach announcements on firm values. *Information & Management*, 46(7):404–410, 2009. 61

[61] S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988. 139

[62] G. Gordon and R. Tibshirani. Karush-kuhn-tucker conditions. *Optimization*, 10(725/36):725. 212

[63] L. A. Gordon and M. P. Loeb. The economics of information security investment. *ACM Transactions in Information System Security*, 5(4):438–457, November 2002. 34, 53, 63

[64] L. A. Gordon, M. P. Loeb, and W. Lucyshyn. Sharing information on computer systems security: An economic analysis. *Journal of Accounting and Public Policy*, 22(6):461–485, 2003. 27, 29, 63, 64

[65] J. Grossklags, N. Christin, and J. Chuang. Secure or insure?: a game-theoretic analysis of information security games. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 209–218, New York, NY, USA, 2008. ACM. 27

[66] J. Grossklags, B. Johnson, and N. Christin. The price of uncertainty in security games. In T. Moore, D. Pym, and C. Ioannidis, editors, *Economics of Information Security and Privacy*, pages 9–32. Springer US, 2010. 27

[67] J. Halpern and V. Teague. Rational secret sharing and multiparty computation: extended abstract. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, STOC '04, pages 623–632, New York, NY, USA, 2004. ACM. 14

[68] J. C. Harsanyi. Games with incomplete information played by "bayesian" players, i-iii. part iii. the basic probability distribution of the game. *Management Science*, 14(7):pp. 486–502, 1968. 16

[69] K. Hausken. Income, interdependence, and substitution effects affecting incentives for security investment. *Journal of Accounting and Public Policy*, 25(6):629 – 665, 2006. 27

[70] K. Hausken. Income, interdependence, and substitution effects affecting incentives for security investment. *Journal of Accounting and Public Policy*, 25(6):629–665, 2006. 29

[71] K. Hausken. Information sharing among firms and cyber attacks. *Journal of Accounting and Public Policy*, 26(6):639–688, 2007. 27, 29, 64

[72] R. Henrion. On constraint qualifications. *Journal of optimization theory and applications*, 72(1):187–197, 1992. 212

[73] IBM. IBM unveils industry's first intelligent cloud security portfolio for global businesses. `https://www-03.ibm.com/press/us/en/pressrelease/45326.wss`. 66

[74] J. Jafarian, E. Al-Shaer, and Q. Duan. Formal approach for route agility against persistent attackers. In J. Crampton, S. Jajodia, and K. Mayes, editors, *Computer Security – ESORICS 2013*, volume 8134 of *Lecture Notes in Computer Science*, pages 237–254. Springer Berlin Heidelberg, 2013. 27

[75] M. Jakobsson and A. Juels. Proofs of work and bread pudding protocols. CMS '99, pages 258–272, Deventer, The Netherlands, 1999. Kluwer, B.V. 149

[76] S. Kakutani. A generalization of brouwer?s fixed point theorem. *Duke mathematical journal*, 8(3):457–459, 1941. 18

[77] S. Karlin. *Mathematical methods and theory in games, programming, and economics*. Addison-Wesley series in statistics. Addison-Wesley Pub. Co., 1959. 34

[78] A. Kashyap, T. Basar, and R. Srikant. Correlated jamming on mimo gaussian fading channels. *Information Theory, IEEE Transactions on*, 50(9):2119 – 2123, sept. 2004. 26

[79] J. Katz and M. Yung. Characterization of security notions for probabilistic private-key encryption. *Journal of Cryptology*, 19(1):67–95, 2006. 139

[80] M. Khouzani, V. Pham, and C. Cid. Incentive engineering for outsourced computation in the face of collusion. In *Proceedings of WEIS 2014 – 13th Annual Workshop on the Economics of Information Security*, 2014. 28, 30

[81] M. Khouzani, V. Pham, and C. Cid. Strategic discovery and sharing of vulnerabilities in competitive environments. In R. Poovendran and W. Saad, editors, *Decision and Game Theory for Security*, volume 8840 of *Lecture Notes in Computer Science*, pages 59–78. Springer International Publishing, 2014. 30

[82] D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, and M. Tambe. Stackelberg vs. nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness. *Journal of Artificial Intelligence Research*, 41:297–327, 2011. 92

[83] H. W. Kuhn. Extensive games and the problem of information. *Annals of Mathematics Studies*, 28, 1953. 16

[84] H. Kunreuther and G. Heal. Interdependent Security. *Journal of Risk and Uncertainty*, 26(2):231–249, March 2007. 27

[85] R. Langner. Stuxnet: Dissecting a cyber warfare weapon. *IEEE Security & Privacy*, 9(3):49–51, 2011. 33

[86] A. Laszka, M. Felegyhazi, and L. Buttyán. A survey of interdependent security games. *CrySyS*, 2, 2012. 27

[87] A. Laszka, G. Horvath, M. Felegyhazi, and L. Buttyan. Flipthem: Modeling targeted attacks with flipit for multiple resources. Technical report, Technical report, Budapest University of Technology and Economics, 2013. 28

[88] B. Laurie and R. Clayton. "proof-of-work" proves not to work. In *IN WEAS 04*, 2004. 191

[89] C. Z. Liu, H. Zafar, and Y. A. Au. Rethinking FS-ISAC: An IT security information sharing network model for the financial services sector. 2013. 27, 64

[90] H. Lovells. DOJ and FTC clarify antitrust implications of cybersecurity information sharing, April 2014. http://www.hoganlovells.com/ [Online; 22-April-2014]. 61

[91] R. Mallik, R. Scholtz, and G. Papavassilopoulos. Analysis of an on-off jamming situation as a dynamic game. *Communications, IEEE Transactions on*, 48(8):1360 –1373, aug 2000. 26

[92] D. P. Mann and J. P. Wissink. Money-back contracts with double moral hazard. *The RAND Journal of Economics*, pages 285–292, 1988. 89

[93] A. Matsui. Information leakage forces cooperation. *Games and Economic Behavior*, 1(1):94–115, 1989. 89

[94] A. Matsui. Cheap-talk and cooperation in a society. *Journal of Economic Theory*, 54(2):245–258, 1991. 89

[95] U. Maurer. Secure multi-party computation made simple. In *Security in Communication Networks*, pages 14–28. Springer, 2003. 13, 125

[96] U. Maurer and S. Wolf. Secret key agreement over a non-authenticated channel — Part I: Definitions and bounds. *IEEE Transactions on Information Theory*, 49(4):822–831, Apr. 2003. 140

[97] C. Miller. The legitimate vulnerability market: Inside the secretive world of 0-day exploit sales. In *In Sixth Workshop on the Economics of Information Security*. Citeseer, 2007. 64

[98] R. A. Miura-Ko, B. Yolken, J. Mitchell, and N. Bambos. Security decision-making among interdependent organizations. In *Computer Security Foundations Symposium, 2008. CSF '08. IEEE 21st*, pages 66–80, June 2008. 27

[99] F. Monrose, P. Wyckoff, and A. D. Rubin. Distributed execution with remote audit. In *NDSS*, 1999. 88

[100] National Institute of Standards and Technology. Technical Guide to Information Security Testing and Assessment. Special Publication 800–115, 2008. 32

[101] National Institute of Standards and Technology. Recommended security controls for federal information systems and organizations. Special Publication 800–53, 2009. 38

[102] Netcraft. Half a million widely trusted websites vulnerable to heartbleed bug, April 2014. news.netcraft.com[Online; 08-April-2014]. 61

[103] J. V. Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944. 15, 17, 18

[104] R. Nix and M. Kantarcioglu. Contractual agreement design for enforcing honesty in cloud outsourcing. In *GameSec*. Springer, 2012. 28, 88, 89

[105] R. Nix and M. Kantarcioglu. Contractual agreement design for enforcing honesty in cloud outsourcing. In *GameSec*. Springer, 2012. 90, 104

[106] A. Nochenson, J. Grossklags, et al. A behavioral investigation of the flipit game. In *12th Workshop on the Economics of Information Security (WEIS)*, 2013. 28, 34

[107] Norton Rose Fullbright. Outsourcing in a brave new world: An international survey of current outsourcing practice and trends. Technical report, Norton Rose Fullbright, 2011. 30, 86

[108] D. of Homeland Security. National cybersecurity and communications integration center. www.us-cert.gov/nccic[Online; Accessed June-2014]. 29, 61

[109] H. Ogut, N. Menon, and S. Raghunathan. Cyber insurance and it security investment: Impact of interdependence risk. In *WEIS*, 2005. 27

[110] C. Orlandi. Is multiparty computation any good in practice? In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5848–5851. IEEE, 2011. 141

[111] R. Pal and P. Hui. Modeling internet security investments: Tackling topological information uncertainty. In J. Baras, J. Katz, and E. Altman, editors, *Decision and Game Theory for Security*, volume 7037 of *Lecture Notes in Computer Science*, pages 239–257. Springer Berlin Heidelberg, 2011. 27

[112] V. Pham and T. Aura. Security analysis of leap of faith protocols. In *SecureComm 2011*, volume 96 of *LNICST*, pages 337–355. Springer-Verlag, 2012. 14, 137

[113] V. Pham and T. Aura. Security analysis of leap of faith protocols. In *SecureComm 2011*, volume 96 of *LNICST*, pages 337–355. Springer-Verlag, 2012. 140

[114] V. Pham and C. Cid. Are we compromised? modelling security assessment games. In *Decision and Game Theory for Security*, pages 234–247. Springer, 2012. 28

[115] V. Pham and C. Cid. Are we compromised? modelling security assessment games. In *GameSec 2012*, volume 7638 of *LNCS*, pages 234–247. Springer-Verlag, 2012. 29

[116] V. Pham, M. Khouzani, and C. Cid. Optimal contracts for outsourced computation. In *Decision and Game Theory for Security*, pages 79–98. Springer, 2014. 30

[117] Press-Release. Government launches information sharing partnership on cyber security, March 2013. www.gov.uk[Online; 27-March-2013]. 61

[118] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 73–85. ACM, 1989. 53

[119] C. Rackoff and D. R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Advances in Cryptology, CRYPTO'91*, pages 433–444. Springer, 1992. 139

[120] T. Radzik. Results and problems in games of timing. *Lecture Notes-Monograph Series*, pages 269–292, 1996. 34

[121] T. Raghavan. Zero-sum two-person games. In R. Aumann and S. Hart, editors, *Handbook of Game Theory with Economic Applications*, volume 2 of *Handbook of Game Theory with Economic Applications*, chapter 20, pages 735–768. Elsevier, 00 1994. 26

[122] E. Rasmusen. Games and information: an introduction to game theory. 1994. 96

[123] M. Raya, M. H. Manshaei, M. Félegyhazi, and J.-P. Hubaux. Revocation games in ephemeral networks. In *Proceedings of the 15th ACM conference on Computer and communications security*, CCS '08, pages 199–210, New York, NY, USA, 2008. ACM. 26

[124] D. Reitter, J. Grossklags, and A. Nochenson. Risk-seeking in a continuous game of timing. In *Proceedings of the 13th International Conference on Cognitive Modeling (ICCM)*, pages 397–403, 2013. 28

[125] R. L. Rivest, A. Shamir, and D. A. Wagner. Time-lock puzzles and timed-release crypto. Technical report, Cambridge, MA, USA, 1996. 220

[126] P. Rogaway. Formalizing human ignorance: Collision-resistant hashing without the keys. In *Progress in Cryptology-VIETCRYPT 2006*, pages 211–228. Springer, 2006. 216

[127] W. Saad, Z. Hart, T. Basar, M. Debbah, and A. Hjørungnes. Physical layer security: coalitional games for distributed cooperation. In *Proceedings of the 7th international conference on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, WiOPT'09, pages 169–176. IEEE Press, 2009. 26

[128] Y. Sagduyu and A. Ephremides. A game-theoretic analysis of denial of service attacks in wireless random access. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops, 2007. WiOpt 2007. 5th International Symposium on*, pages 1 –10, april 2007. 26

[129] B. Schneier. The process of security. `https://www.schneier.com/essays/archives/2000/04/the_process_of_secur.html`. 65

[130] J. Seth. Strategic importance of information technology. *Advances in Telecommunications Management*, pages 3–16, 1994. 12

[131] S. Setty, R. McPherson, A. J. Blumberg, and M. Walfish. Making argument systems for outsourced computation practical (sometimes). In *NDSS*, 2012. 28, 30, 88

[132] S. Setty, R. McPherson, A. J. Blumberg, and M. Walfish. Making argument systems for outsourced computation practical (sometimes). In *NDSS*, 2012. 30, 86

[133] S. Setty, V. Vu, N. Panpalia, B. Braun, A. J. Blumberg, and M. Walfish. Taking proof-based verified computation a few steps closer to practicality. In *USENIX Security*, 2012. 28, 30, 88

[134] S. Setty, V. Vu, N. Panpalia, B. Braun, A. J. Blumberg, and M. Walfish. Taking proof-based verified computation a few steps closer to practicality. In *USENIX Security*, 2012. 30, 86

[135] A. Shamir. How to share a secret. *Commun. ACM*, 22:612–613, November 1979. 13

[136] C. Shapiro. Exchange of cost information in oligopoly. *The review of economic studies*, 53(3):433–446, 1986. 29, 63

[137] E. Solan and L. Yariv. Games with espionage. *Games and Economic Behavior*, 47(1):172–199, 2004. 89

[138] T. Spyridopoulos, G. Karanikas, T. Tryfonas, and G. Oikonomou. A game theoretic defence framework against dos/ddos cyber attacks. *Computers & Security*, 38(0):39 – 50, 2013. Cybercrime in the Digital Economy. 27

[139] T. Tryfonas. On security metaphors and how they shape the emerging practice of secure information systems development. *Journal of Information System Security*, 3(3):21–50, 2007. 63

[140] USC. US code: Title 44, section 3542, Jan 2012. 13

[141] M. van Dijk, A. Juels, A. Oprea, and R. L. Rivest. Flipit: The game of "stealthy takeover". Cryptology ePrint Archive, Report 2012/103, 2012. 27, 28, 32, 33, 34, 35, 36, 37, 38, 43, 46, 196

[142] J. Vijayan. Breach, undetected since '05, exposes data on Kingston customers, July 17, 2007. `http://www.computerworld.com/s/article/9027220/Breach_undetected_since_05_exposes_data_on_Kingston_customers`. 33

[143] X. Vives. Trade association disclosure rules, incentives to share information, and welfare. *RAND Journal of Economics*, 21(3):409–430, 1990. 29, 63

[144] C. Wang, K. Ren, and J. Wang. Secure and practical outsourcing of linear programming in cloud computing. In *INFOCOM, 2011*, 2011. 88

[145] B. Waters, A. Juels, J. A. Halderman, and E. W. Felten. New client puzzle outsourcing techniques for dos resistance. ACM CCS '04, pages 246–256, New York, NY, USA, 2004. ACM. 149

[146] F.-L. Wong, F. Stajano, and J. Clulow. Repairing the bluetooth pairing protocol. In B. Christianson, B. Crispo, J. Malcolm, and M. Roe, editors, *Security Protocols*, volume 4631 of *Lecture Notes in Computer Science*, pages 31–45. Springer Berlin Heidelberg, 2007. 14

[147] Q. Xiong and X. Chen. Incentive mechanism design based on repeated game theory in security information sharing. In *2nd International Conference on Science and Social Research (ICSSR 2013)*. Atlantis Press, 2013. 64

[148] G. Yan and S. Eidenbenz. Ddos mitigation in non-cooperative environments. In A. Das, H. Pung, F. Lee, and L. Wong, editors, *NETWORKING 2008 Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*, volume 4982 of *Lecture Notes in Computer Science*, pages 599–611. Springer Berlin Heidelberg, 2008. 27

[149] K. Yi, F. Li, G. Cormode, M. Hadjieleftheriou, G. Kollios, and D. Srivastava. Small synopses for group-by query verification on outsourced data streams. *ACM TODS*, 2009. 88

[150] C. Zhou, S. Karunasekera, and C. Leckie. A peer-to-peer collaborative intrusion detection system. In *Networks, 2005. Jointly held with the 2005 IEEE 7th Malaysia International Conference on Communication., 2005 13th IEEE International Conference on*, volume 1, page 6 pp., nov. 2005. 26

[151] Q. Zhu and T. Basar. Dynamic policy-based ids configuration. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 8600 –8605, dec. 2009. 26

[152] Q. Zhu, C. Fung, R. Boutaba, and T. Basar. A game-theoretical approach to incentive design in collaborative intrusion detection networks. In *Game Theory*

for Networks, 2009. GameNets '09. International Conference on, pages 384 –392, may 2009. 27

[153] Q. Zhu, H. Li, Z. Han, and T. Baş andar. A stochastic game model for jamming in multi-channel cognitive radio systems. In *Communications (ICC), 2010 IEEE International Conference on*, pages 1 –6, may 2010. 26

[154] M. Zviran and W. J. Haga. Password security: an empirical study. *Journal of Management Information Systems*, pages 161–185, 1999. 63

Appendix A

Basic of Karush-Kuhn-Tucker (KKT) Optimisation

Named after its developers William Karush, Harold W. Kuhn, and Albert W. Tucker, the KKT conditions provide an essential tool for non-linear optimisation. It can be stated in the following [62]. Consider the following optimisation problem:

$$\begin{aligned} \min_{\boldsymbol{x}} \ & f(\boldsymbol{x}) \\ \text{subject to: } & g_i(\boldsymbol{x}) \leq 0 \ \forall i \in [n], and, \\ & h_j(\boldsymbol{x}) = 0 \ \forall j \in [m], \end{aligned}$$

for functions $f, g_i, h_j : \mathbb{R}^k \to \mathbb{R}$ and some $n, m, k \in \mathbb{Z}^+$. Suppose that $f, g_i.h_j$ for $i \in [n]$, $j \in [m]$ are continuously differentiable in $\boldsymbol{x}^* \in \mathbb{R}^k$, then if $\boldsymbol{x}^*$ is a local minimum of the above problem and that it satisfies a *regularity condition*, there exist constants μ_i for $i \in [n]$, and λ_j for $j \in [m]$ such that the following KKT conditions hold

$$\begin{aligned} \text{Stationary: } & \nabla f(\boldsymbol{x}^*) + \sum_{i=1}^{n} \mu_i \nabla g_i(\boldsymbol{x}^*) + \sum_{j=1}^{m} \lambda_j \nabla h_j(\boldsymbol{x}^*) = 0 \\ \text{Complementary slackness: } & \mu_i g_i(\boldsymbol{x}^*) \ \forall \ i \in [n] \\ \text{Primal feasibility: } & g_i(\boldsymbol{x}^*) \leq 0 \ \forall \ i \in [n], \ h_j(\boldsymbol{x}^*) = 0 \ \forall \ j \in [m] \\ \text{Dual feasibility: } & \mu_i \geq 0 \ \forall \ i \in [n] \end{aligned}$$

There are many regularity conditions that $\boldsymbol{x}^*$ can satisfy in order for it to qualify the KKT conditions above. For the purpose of our work, we only mention the so-called Mangasarian-Fromovitz constraint qualification (MFCQ) [72], i.e., the following vectors

are positive-linearly independent in $\mathbb{R}^k$:

$$\nabla h_j(\boldsymbol{x}^*) \ \forall \ j \in [m], \ \nabla g_i(\boldsymbol{x}^*) \ \forall \ i \in [n] \text{ s.t. } g_i(\boldsymbol{x}^*) = 0$$

Our process of optimising outsourcing contracts proceed as follows. We first prove that the cost function to minimise f, along with g_i (there is no h_j) satisfies MFCQ regularity condition for all non-trivial points $\boldsymbol{x}^*$. This ensures that all interesting local minima of f will satisfy the KKT conditions. The remainder of the work is to write a program that solves KKT conditions to find all points that satisfy them, then comparing the found results to eliminate points which are not local minima. The remaining points thus form the global minima of f.

Appendix B

Mathematica Code for KKT Optimisation

The program takes as input several parameters, including $eqsys$, var, sup, and $supVar$. It means to solve the following optimisation, for a tuple of k variables $\boldsymbol{x}$ and n constraints:

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \text{ subject to } g_i(\boldsymbol{x}) \leq 0 \ \forall i \in [n].$$

To do so, set $eqsys = \{f(\boldsymbol{x}), g_1(\boldsymbol{x}), \ldots, g_n(\boldsymbol{x})\}$, $var = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k\}$. The other parameter sup is of the form $\{\text{inequality } 1, \ldots, \text{inequality t}\}$, which gives further restrictions to the solution. For example in Section 4.5 we need to optimise two-agent contracts, in which there are extra requirements such as $\Lambda < 1$, $\gamma > 0$, etc. These restrictions would later be used to filter optimal solutions that are not meaningful, e.g., if it assumes that $c < 0$, or $\Lambda > 1$. Finally, $supVar$ gives the list of parameters that appear in sup, e.g., c, Λ, γ.

Listing B.1: Optimisation using KKT conditions

```
Clear[KKT];
KKT[eqsys_, var_, sup_, supVar_] := (
  min = eqsys [[1]];
  len = Length[eqsys] - 1;
  nMax = 2^(Length[eqsys] - 1);
  tmpEqSys =
   Table[eqsys [[ kh ]] Subscript[\[Mu], kh - 1], {kh, 1,
     len + 1}] /. {Subscript[\[Mu], 0] -> 1};
  orgSys =
   Table[D[Plus @@ tmpEqSys, var[[kh]]] == 0, {kh, 1, Length[var ]}];
  For[ i = 0, i < nMax, i++, (
```

```
12      sys = orgSys;
13      newEqSys = {};
14      For[j = 0, j < len, j++, (
15        sys =
16         Union[sys,
17          If[Mod[BitShiftRight[i, j], 2] ==
18            0, {Subscript[\[Mu], j + 1] == 0}, {eqsys[[j + 2]] == 0}]];
19        newEqSys =
20         Union[newEqSys,
21          If[Mod[BitShiftRight[i, j], 2] ==
22            0, {eqsys[[j + 2]] <= 0}, {eqsys[[j + 2]] == 0}]];
23        )];
24      sol =
25       Solve[sys,
26        Union[var, Table[Subscript[\[Mu], kh], {kh, 1, len}]]];
27      If[sol == {}, , (
28        \[Mu]Conditions =
29         Table[Subscript[\[Mu], kh] >= 0, {kh, 1, len}];
30        For[mh = 1, mh <= Length[sol], mh++,
31         (Print[sol[[mh]]]; Print["To reduce:"];
32          Print[And @@
33            Flatten[
34             Union[sup /. sol[[mh]],
35              newEqSys /. sol[[mh]], \[Mu]Conditions /. sol[[mh]]]]];
36          Print["Start reducing ..."];
37          abc = Reduce[
38            And @@ Flatten[
39             Union[sup /. sol[[mh]],
40              newEqSys /. sol[[mh]], \[Mu]Conditions /. sol[[mh]]]],
41            supVar];
42          If[abc == False, Continue[]];
43          Print["Solution = ", sol[[mh]] // FullSimplify, "\n",
44           "\nConditions = ",
45           And @@ Flatten[
46             Union[newEqSys /. sol[[mh]],
47              orgSys /. sol[[mh]], \[Mu]Conditions /. sol[[mh]]]] //
48            FullSimplify, "\n\n"];)
49         ]
50        )];
51       );
52     ];) ;
```

Appendix C

A Key-Exchange Protocol for Definition 1.16

In order for our schemes to work, we need the Decisional Diffie-Hellman (DDH) assumption and the existence of a collision-resistant hash function, defined below:

Definition C.1 ([25])**.** *Consider a generator $\mathcal{G}$ that on input n would output a cyclic group $\mathbb{G}$ of prime order q and generator g. We say that the* DDH problem is hard relative to $\mathcal{G}$ *if for all PPT algorithms $\mathcal{A}$ there exists a negligible function ϵ such that*

$$|\Pr[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^{xy}) = 1] - \Pr[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^z) = 1]| \leq \epsilon(n)$$

where the probabilities are taken over $\mathcal{G}(n)$, and the uniform randomness of $x, y, z \in \mathbb{Z}_q$.

Definition C.2 ([126])**.** *A tuple (Gen, H) where* Gen *is a key generator, and H is a hash function, is said to be a* collision-resistant hash function *if for all PPT algorithms $\mathcal{A}$ there exists a negligible function such that*

$$\Pr[k \leftarrow \mathsf{Gen}(n); (m, m') \leftarrow \mathcal{A}(k, n) : m \neq m' \wedge H(k, m) = H(k, m')] \leq \epsilon(n).$$

The existence of the desired key-exchange protocol is stated in the following result:

Proposition C.1. *Assume that the DDH problem is hard relative to some generator $\mathcal{G}$, and that there exist a collision-resistant hash function (Gen, H) and a non-malleable commitment scheme $(\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open})$, then under the common-reference string (CRS) model there exists a key-exchange protocol (I, R) satisfying the correctness and key indistinguishability properties of Definition 1.16. Assume that there exists an authenticated encryption scheme $(\mathcal{K}, \mathsf{Enc}, \mathsf{Dec})$ where keys are generated by honest executions of (I, R), then there exists a protocol $(\mathsf{I}', \mathsf{R}')$ satisfying Definition 1.16.*

Proof. With the CRS assumption, we first assume that there exists a *trusted third party* that generates $(\mathbb{G}, g, q) \leftarrow \mathcal{G}(n)$, $k \leftarrow \mathsf{Gen}(n)$ and $\mathsf{CK} \leftarrow \mathsf{Setup}(n)$, and distribute these to the two parties performing key exchange. Then, we construct the protocol (I, R) as follows:

1. I: generates $x \leftarrow_\$ \mathbb{Z}_q$, $g_I = g^x$, $(c_I, d_I) \leftarrow \mathsf{Commit}_{\mathsf{CK}}(g_I)$, and sends c_I.
2. R: generates $y \leftarrow_\$ \mathbb{Z}_q$, $g_R = g^y$, $(c_R, d_R) \leftarrow \mathsf{Commit}_{\mathsf{CK}}(g_R)$, and sends c_R.
3. I: receives c_R and sends d_I.
4. R: receives d_I and sends d_R.
5. I: computes $g_R \leftarrow \mathsf{Open}_{\mathsf{CK}}(c_R, d_R)$, if $g_R = \perp$, returns $\perp$, otherwise computes $k_{\mathsf{sess}} \leftarrow H(k, g_R^x)$.
6. R: computes $g_I \leftarrow \mathsf{Open}_{\mathsf{CK}}(c_I, d_I)$, if $g_I = \perp$, returns $\perp$, otherwise computes $k_{\mathsf{sess}} \leftarrow H(k, g_I^y)$.

The correctness of this mechanism with respect to Definition 1.16 is straightforward,

$$k_{\mathsf{I}} = H(k, g_{\mathsf{R}}^x) = H(k, g^{xy}) = H(k, g_{\mathsf{I}}^y) = k_{\mathsf{R}}.$$

For key indistinguishability, we first assume for some adversary Adv there exists a non-negligible function ϵ' such that

$$\Pr[(k_0, \cdot, k_1) \leftarrow (\mathsf{I}(n), \mathsf{Adv}, \mathsf{R}(n)) : k_0 = k_1 \neq \perp] = \epsilon'(n)$$

Let g'_R and g'_I be the Diffie-Hellman components received by I and R, respectively. Given that $k_0 = k_1$, two situations might have occurred:

- $g'^x_R \neq g'^y_I$: this means that $H(k, g'^x_R) = k_0 = k_1 = H(k, g'^y_R)$, which only occurs with negligible probability due to the collision property of hash function.
- $g'^x_R = g'^y_I$: Let tr_I and tr_R be the message transcripts of the communication perceived by I and R, respectively. Consider two sub-cases:
  - $\mathsf{tr}_I = \mathsf{tr}_R$: this indicates that Adv did not modify any message at all. Therefore, it receives g^x, g^y, and that $k_0 = k_1 = g^{xy}$. The key indistinguishability thus becomes the DDH experiment, and thus the adversary's success is at most $1/2$ plus some negligible probability.

– $\mathsf{tr}_I \neq \mathsf{tr}_R$: Let (c'_I, d'_I) and (c'_R, d'_R) be commitment values received by R and I, respectively. If either $c'_I = c_I$ or $c'_R = c_R$, then due to the binding property of commitment, with at most negligible probability the adversary can make both $g_I \neq g'_I$ and $g_R \neq g'_R$. Otherwise we have either $g_I = g'_I$ or $g_R = g'_R$. This implies that $k_0 = k_1 = g^{xy}$, and thus the key indistinguishably again becomes the DDH experiment, which implies that the adversary's success is at most $1/2$ plus some negligible probability.

  Consider the case $c_I \neq c'_I$ and $c_R \neq c'_R$. From the construction it is easy to see that either the adversary does not receive d_I before producing c'_I, or it does not receive d_R before producing c'_R. Assume the former holds, due to the non-malleability property of commitment, the value of g_I and g'_I are statistically independent, and so are x and x' which they respectively correspond to. Due to the hiding property, g_I and g'_R are also statistically independent, and so are x and y'. Therefore, we eventually have $k_0 = g^{xy'}$ is statistically indistinguishable from some g^c for $c \leftarrow_\$ \mathbb{Z}_q$, and that it is also statistically independent from $k_1 = g^{x'y}$. This means that the probability that $k_0 = k_1$ is negligible. A similar analysis also applies when we consider that fact that the adversary does not receive d_R before producing c'_R.

The above points together imply that the adversary's success probability in the key indistinguishability is negligibly different from $1/2$. We now construct $(\mathsf{I}', \mathsf{R}')$ from (I, R) that satisfies also the synchronisation property. Indeed, $(\mathsf{I}', \mathsf{R}')$ inherits all six steps of (I, R), along with the following additions:

7. I': produces $c_I \leftarrow \mathsf{Enc}_{k_{\mathsf{sess}}}(\mathsf{tr}_I)$, where tr_I is the transcript of all previous messages, and sends c_I.

8. R': produces $c_R \leftarrow \mathsf{Enc}_{k_{\mathsf{sess}}}(\mathsf{tr}_R)$, where tr_I is the transcript of all previous messages, and sends c_R.

9. I': computes $\mathsf{tr}_R \leftarrow \mathsf{Dec}_{k_{\mathsf{sess}}}(c_R)$, and if $\mathsf{tr}_R \neq \mathsf{tr}_I$, then set $k_{\mathsf{sess}} = \perp$.

10. I': computes $\mathsf{tr}_I \leftarrow \mathsf{Dec}_{k_{\mathsf{sess}}}(c_I)$, and if $\mathsf{tr}_I \neq \mathsf{tr}_R$, then set $k_{\mathsf{sess}} = \perp$.

We note that $(\mathsf{I}', \mathsf{R}')$ preserves the correctness and key indistinguishability properties because the last four steps above do not have a possibility to change the value of k_{sess} to anything other than $\perp$. The synchronisation property holds because if $k_0 = k_1$ and $\mathsf{tr}_I \neq \mathsf{tr}_R$, due to Lemma 5.7 the adversary is able to produce valid ciphertexts that pass the checks in step 9 and 10 above with negligible probability, that is, $k_0 = k_1 \neq \perp$ with negligible probability. Otherwise, if $\mathsf{tr}_I \neq \mathsf{tr}_R$ the adversary's only mean of modifying

the communication is to produce either $c'_I \neq c_I$ or $c'_R \neq c_R$ in steps 7 or 8, respectively. However, the IND-CTXT property of encryption guarantees that these modifications are accepted with negligible probability, i.e., they pass test 9 and 10 with negligible chances, which also implies $k_0 = k_1 \neq \perp$ occurs with negligible probability. This completes the proof of security of key exchange. □

Appendix D

A Proof-of-Work Scheme for Definition 1.17

To satisfy Definition 1.17, we adapt the constant-time puzzle scheme proposed by Rivest et al. [125]. First, we assume the existence a generator ModGen that on integer input $n > 0$ outputs a pair of primes p, q of n-bit length, as well as a family of hash functions (Gen, H_m) for all $m \in \mathbb{Z}^+$ that maps the message space $\{0,1\}^*$ to hash space $\mathbb{Z}_m^* = \mathbb{Z}_m \setminus \{0,1\}$. Then suppose there exists a trusted third party that generates $k \leftarrow \mathsf{Gen}(n)$ and makes k publicly and securely accessible. The $(\mathsf{Prove}, \mathsf{Verify})$ proceeds as follows for integer cost c:

1. Verify: generates $(p, q) \leftarrow \mathsf{ModGen}(n)$ and sends pq.
2. Prove: receives m, computes $g \leftarrow H_{pq}(k, id)$, $t \equiv g^{2^c} \pmod{m}$, and sends t.
3. Verify: receives t, computes $g \leftarrow H_{pq}(k, id)$, $e \equiv 2^c \pmod{(p-1)(q-1)}$, $t' \equiv g^e \pmod{pq}$, and returns true if $t = t'$ or $\perp$ otherwise.

When cost c is non-integer, the prover and the verifier could engage in a proof-of-work with cost $\lfloor c \rfloor$, and for the remainder $r = c - \lfloor c \rfloor$ the prover could ask the verifier to perform any simple costly computation, e.g., hashing the id repeatedly multiple times until the observed cost matches r. Nevertheless, the security of this proof-of-work mechanism is provided below, followed by a remark on the practicality of (D.1):

Proposition D.1. *Assume that the cost of squaring in modular arithmetic is one (unit cost), and that any computation with cost $p \in [0,1]$ would give the correct squaring result with probability at most p. In other words, assume that for all PPT algorithms $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, all positive integer $c > 0$, all $g \in \mathbb{Z}_{pq}^*$, and all probabilities p, there exists*

a negligible function ϵ such that

$$\Pr\left[\begin{array}{c}(p,q) \leftarrow \mathsf{ModGen}(n), s \leftarrow \mathcal{A}_1(g,c), t \leftarrow \mathcal{A}_2^{\mathsf{sq}(\cdot,\cdot)}(s,g,c,pq) : \\ \mathsf{cost}(\mathcal{A}) \leq c \cdot p \wedge t \equiv g^{2^c} \pmod{pq}\end{array}\right] \leq p + \epsilon(n) \qquad \text{(D.1)}$$

where sq *on input* g, m *would output* $g^2 \pmod m$*,* $\mathcal{A}$ *is not allowed to query* sq *with inputs of the form* $(g, \cdot)$*, and that* $\mathsf{cost}(\mathcal{A})$ *ignores the computation cost of* sq*. Assume that there exists a family of collision-resistant hash functions* (Gen, H_m) *for all* $m \in \mathbb{Z}^+$*, then the proof-of-work scheme above satisfies Definition 1.17 for all integer costs* $c > 0$*.*

Proof. As usual, the proof of correctness is rather trivial. We notice the totient function $\phi(pq) = (p-1)(q-1)$, therefore

$$t \equiv g^{2^c} \equiv g^{2^c \pmod{\phi(pq)}} \equiv g^e \equiv t' (\text{mod } pq)$$

and thus Verify would return true. On the other hand, the computation of $g^{2^c} \pmod{pq}$ requires c successive squaring computations, which bears the cost of c. This thus proves the correctness property of proof-of-work.

For the verifiability property, due to the collision-resistance property of hash functions, $\mathcal{A}$ can find $id' \neq id$ such that $g = H_m(k, id) = H_m(k, id') = g'$ with negligible probability. Otherwise, the fact that $\mathsf{Adv2}$ cannot query Prove with id matches that $\mathcal{A}$ cannot ask sq to square $g = H_m(k, id)$ with respect to any modulus. Meanwhile, the adversary Adv_1 is given id and c matches that $\mathcal{A}_1$ is given c and g. Thus, the verifiability experiment becomes the squaring experiment as described in (D.1). Therefore, given that (D.1) holds, the adversary $\mathsf{Adv} = (\mathsf{Adv}_1, \mathsf{Adv}_2)$ succeeds with probability at most $p + \epsilon(n)$. □

Remark. The assumption expressed in (D.1) is made possible by several observations. First, $\mathcal{A}_1$ represents the adversary's preparation before engaging in the process of proving its proof-of-work, and therefore it should not be given pq, which is generated freshly by the verifier Verify. This can be strengthened by designing ModGen that generates p and q at random. Meanwhile, the fact that $\mathsf{cost}(\mathsf{A}_2) \leq c \times p$ indicates a success probability $p + \epsilon(n)$ results from the fact that it is hard to reduce the number of squaring computations, and that any attempt to lessen the code of each squaring would reduce accuracy of the result. The former is ensured by the assumption of hardness in integer factorisation: the chance for any adversary to factor pq for n-bit length primes p and q is at most $\epsilon(n)$. Indeed, the only known effective way to lessen the number of squaring computations is to compute $2^c \pmod{\phi(pq)}$, which is possible only

if knowledge of p and q are known. For the latter, we notice that during squaring modulo pq, some steps can be omitted, for example the computation of the last few bits of the remainder. In such case although the reduce in computation cost is small, it is nevertheless non-negligible. However, this means that the adversary must guess the unknown part of the result, and may succeeds with sharply reduced chance. We thus reasonably assume that the fraction of save in cost is always less than the reduce in accuracy, as in (D.1).