Rational authentication protocols

Long Hoang Nguyen
Oxford University Department of Computer Science
Parks Road, OX1 3QD, Oxford, England
Email: Long.Nguyen@cs.ox.ac.uk

ABSTRACT

We use ideas from game theory to improve two families of authentication protocols, namely password-based and manual authentication schemes. The protocols will be transformed so that even if an intruder attacks different protocol runs between honest nodes, its expected payoff will still be lower than when it does not attack. A *rational* intruder, who always tries to maximise its payoff, therefore has no incentive to attack any protocol run among trustworthy parties.

1. INTRODUCTION

Ideas from game theory have been used to re-design a number of fair exchange protocols [3, 20] and secret sharing schemes [8, 6] so that nodes cannot act on their own interests to bring these schemes to failure. As an example, in a fair exchange, a party accepts to deliver an item iff it receives another item in return, and hence even unmalicious but self-interested parties will be tempted to deviate from a protocol to gain advantage. This notion of players' rationality or self-interest is however not applicable to authentication and key-agreement protocols where all honest nodes should cooperate to complete a protocol successfully, because it is in their mutual interest that they agree on the same data.

We instead observe that in many environments, e.g. the financial industry, the intruder can be *rational* in the sense that it always tries to maximise its payoff as in the following scenario. If the intruder has somehow obtained A's financial details such as bank statement, it will know that A can have a large amount of money. This could imply that the potential reward of having access to A's account exceeds the cost of launching many attacks on different protocol runs. The intruder is therefore highly motivated to attack the authentication stage of online transactions carried out between A and the bank.[1] The observations motivate us to use techniques in game theory to redesign authentication protocols to resist this kind of rational intruder.

Our first contribution presented in Section 2 is a general protocol transformation that is applicable to a variety of authentication protocols. In this transformation an honest node, who is usually the protocol initiator, will pursue some additional behaviour under some probability after each successful protocol session. The combination of the behaviour and its occurrence probability is designed to ensure that an intruder's expected payoff in attacking the protocol is lower than its expected payoff in not attacking. The intruder therefore does not have any incentive to misbehave. Since the additional behaviours of the initiator must benefit the intruder, they vary from one to another applications but an example with respect to the above banking scenario can be given as follows. To avoid being disrupted, the account holder A can occasionally make a tiny payment to a third party who is the intruder in disguise after each successful transaction. The questions we therefore want to answer are: How much is the tiny payment? and How often does A need to make such a payment to successfully discourage the intruder from attacking?

The main thrust of this paper is to formally demonstrate how this protocol transformation works and benefits two families of pairwise authentication protocols. They are password based authentication schemes of Section 3 and manual authentication protocols of Section 4, though other cryptographic protocols such as distance-bounding schemes [5, 13] have also been demonstrated to benefit from our work.[2] Throughout the sections, we will largely abstract away from the exact detail of additional behaviours which are not immediately important to our analysis until Section 5.

To assess the performance of our analysis, in Section 5 we present a case study on the above scenario. While we will use our results derived in Section 3 to answer questions posed previously, our analysis will also shed new light on the usability and economic security of current banking applications regarding the limit of number of consecutive failed attempts of entering the correct password.

In Section 6, we show how the protocol transformation can be adapted to work with group protocols. In particular, the *Machiavelli* adversary model of Syverson et al. [20] will become useful in our analysis of group protocols based on passwords where compromised nodes do not share secret with the intruder.

Our use of additional behaviours in honest parties' activities tailored for authentication protocols can be traced back to earlier work in other context of rational secret sharing schemes. To encourage an intruder to give up attacking

9[1] Although a limit is usually put on the number of consecutive failed attacks, the intruder can still launch an attack whenever A carries out an online transaction until A's account is blocked. Consequently A will not be able to do trading with any others.

9[2] Our work reported here was first posted on the IACR Cryptology ePrint Archive [14], and subsequently influenced the author's work in the design of rational distance-bounding protocols [13]. A distance-bounding protocol relies on a rapid-bit exchange, which therefore makes it very different to deal with compared to password based and manual authentication schemes considered here.

Protocol transformation

The protocol initiator A pursues the following strategy to discourage a rational intruder from attacking protocol runs of honest parties.

Upon each successful protocol session, which happens when the intruder either does not interfere with or succeeds in its attack on the protocol.

- With probability $\alpha \in [0, 1)$: the initiator A is *generous* and will pursue some additional behaviour that benefits the intruder. The exact additional behaviour depends on the intruder's goals in different scenarios, but a simple example might be as follows. To avoid being disrupted, an account holder A donates a tiny amount of money to a third party who is the intruder in disguise after each successful transaction. More details can be found in our case study of Section 5.

  The intruder will get payoff U when it does not attack or U_1^+ when it successfully attacks the protocol.

- With probability $1-\alpha$: A is ungenerous and pursues no further activity. There is no payoff for the intruder if it behaves honestly, but if the intruder attacks and succeeds it will still get a payoff U_2^+.

Upon each unsuccessful protocol session, which usually happens when the intruder fails in its attack. The initiator A will not pursue any additional behaviour, and the intruder receives a negative payoff U^- due to, e.g., the cost of launching an attack on a protocol run.

Since the intruder much benefits from a successful attack regardless of whether A is generous or not, we arrive at

$$min\{U_1^+, U_2^+\} > U > 0 > U^-$$

Table 1: Protocol transformation.

Strategy of intruder	Protocol session	Strategy of initiator	Payoff of intruder
No attack	Succeed	Ungenerous	0
No attack	Succeed	Generous	U
Attack	Succeed	Ungenerous	U_1^+
Attack	Succeed	Generous	U_2^+
Attack	Fail	Ungenerous	U^-
The lower half is the worst case scenario of the upper half.			
No attack	Succeed	Ungenerous	0
No attack	Succeed	Generous	U
Attack	Succeed	Any	$U^+ = max\{U_1^+, U_2^+\}$
Attack	Fail	Ungenerous	U^-

Table 2: A summary of the game.

a protocol, it is probably inevitable that we need to give something, which is less damaging than a successful attack, to the intruder in each normal run. Both rational secret sharing schemes of Gordon and Katz [8] and Fuchsbauer et al. [6] follow this strategy by allowing a trusted dealer to send invalid shares of secret to players at the beginning of some iterations, or forcing nodes to proceed in a sequence of fake runs followed by a single real one. In addition we further realise relationship between our work and the theory of utility function in economics that underlies the business of insurance and lottery. The latter will be discussed in our conclusion of this paper.

2. PROTOCOL TRANSFORMATION

For simplicity pairwise authentication schemes are considered, where two parties A and B want to authenticate or agree on the same data. In the schemes, it is in honest nodes' mutual interest that they follow the protocol. Among the protocol participants, there is one party who initiates a protocol by, e.g., sending the first message and hence we denote A the protocol *initiator*. No specific protocol is given until multiple-run attacks are considered in subsequent sections, because for single-run attacks our suggested changes in the behaviour of the initiator A are independent of the type of authentication protocols whether they are based on passwords [1] or human interactions [11, 22]. Our analysis will be generalised to deal with group authentication scenarios in Section 6.

Prior to proceeding to the next paragraph, we would strongly recommend the readers to study the protocol transformation provided in Tables 1 and 2, which also introduce the notation for the intruder's payoffs, i.e. U, U_1^+, U_2^+, U^-, with different combination of parties' strategies and protocol outcomes. The payoffs, which are often quantified in terms of money, depend on a number of factors, including the cost of launching attacks (computation or energy consumption) and financial reward of a successful attempt. As in many rational secret sharing schemes introduced to date [8, 6], we assume here that the payoffs are known to both protocol participants and the intruder in advance. Moreover our analysis in this section as well as Sections 3 and 4 does not require us to specify the additional behaviour of the initiator in the protocol transformation, because its abstract form in terms of the corresponding payoff for the intruder is sufficient. We will provide justification for the assumptions when a case study is provided in Section 5.

From Tables 1 and 2, we observe that the difference between the payoffs U_1^+ and U_2^+ for an attacking intruder can vary, e.g. U_1^+ and U_2^+ can be either far apart or roughly the same. We therefore will only tackle the worst case scenario here: regardless of whether A is generous or not the intruder's payoff is $U^+ = max\{U_1^+, U_2^+\}$ when it launches a successful attack as seen in the lower half of Table 2. A solution for the worst case scenario applies to every other scenario where $U_1^+ \neq U_2^+$.

Using the protocol transformation specified in Table 1, we arrive at the following theorem.

Theorem 1. If an intruder can only attack up to a single run of an authentication protocol and succeed with probability ϵ, then to discourage the intruder from attacking protocol runs between honest nodes, this inequality must

hold:

$$\alpha > \frac{\epsilon U^+ + (1-\epsilon)U^-}{U}$$

PROOF. If the intruder does not misbehave, his expected payoff in each run is αU. If the intruder misbehaves, his expected payoff of a single-run attack is $\epsilon U^+ + (1-\epsilon)U^-$.

So as long as $\alpha U > \epsilon U^+ + (1-\epsilon)U^-$ or $\alpha > [\epsilon U^+ + (1-\epsilon)U^-]/U$, it is in the intruder's interest not to attack any protocol runs of honest nodes. □

Although U^+ is usually significantly bigger than U, ϵU^+ can still turn out to be less than U. This can be done by, e.g., choosing a password of a reasonable length so that the probability of a successful attack ϵ is small.

Suppose that there are a number of strategies regarding different values of α that node A can pursue, then α is selected big enough to meet the requirements of Theorem 1. In other words, α is big enough that the intruder's expected payoff is higher if it behaves honestly, as honest parties always prefer not to give the intruder too much benefit.

The above analysis only takes into account single-run attacks, in practice a rational intruder as defined in Section 1 would attack multiple protocol runs. For this reason, it is desirable that we consider the case of multiple-run attacks on authentication protocols.

3. MULTIPLE-RUN ATTACKS ON PASSWORD-BASED PROTOCOLS

Any secure password-based (authentication or key-agreement) protocol usually need to resist off-line searching, i.e. the only way to find out a guess of a password is correct is to interact with the protocol participants.[3] Our analysis here applies to a variety of password-based protocols, but for clarity we give the definition of the Diffie-Hellman-based Encrypted Key Exchange scheme of Bellovin and Merritt [1]. This protocol establishes a shared private key g^{xy}, where g^x and g^y are Diffie-Hellman keys of A and B, from a short password pw using an encryption scheme $E_{pw}()$ and a cryptographic hash function $hash()$.

Since passwords are usually short and unchanged for a period of time, the chance of a successful attack increases quite significantly as more and more attempts are launched to guess the passwords. We stress that this feature of a password-based scheme, which is different from other kinds of authentication protocol, is particularly relevant to our discussion, because it will encourage a rational intruder to keep attacking and guessing the password in multiple protocol runs.

Encrypted Key Exchange Protocol [1]
1. $A \longrightarrow B \ : \ A \parallel E_{pw}(g^x)$
2. $B \longrightarrow A \ : \ E_{pw}(g^y) \parallel hash(sk \parallel 1)$
where $sk = hash(A \parallel B \parallel g^x \parallel g^y \parallel g^{xy})$
3. $A \longrightarrow B \ : \ hash(sk \parallel 2)$

In practice we usually limit the number of failed attempts an intruder can make, e.g. three wrong guesses and the protocol will stop running, and thus we denote k the limit of number of attacks an intruder can launch on a protocol.

In the simplest scenario, passwords are uniformly and randomly selected from $\{1,\ldots,n\}$, then[4] $1 \le k \le n$ and the chance of correctly guessing the password the first time is $\epsilon = \epsilon_1 = 1/n$. If the first guess is incorrect, then the second guess succeeds with probability $\epsilon_2 = 1/(n-1)$. For all $k \in \{1,\ldots,n\}$ we have $\epsilon_k = 1/(n-k+1)$. Our analysis can be adapted to deal with non-uniformly distributed passwords when the distribution of passwords is known to the intruder and all values of password are chosen independently, i.e. the conditional probability ϵ_k does not depend on which $k-1$ passwords have been tried beforehand.

In order to be precise in our arguments, we need to be clear about the attacking strategy of the intruder that our protocol transformation of Table 1 seeks to resist. If the intruder decides to attack a protocol up to k runs, then the intruder only terminates its attack if either of the following two conditions is met:

- The intruder succeeds in the t^{th} attempt where $t \le k$ or
- The intruder fails in all k attempts.

These k attempts do not need to be consecutive and can be interleaved with any number of protocol runs which are not attacked by the intruder.

In many circumstances this is the optimal strategy for any intruder who decides to attack. For example, A is running an authentication protocol with a bank where A has an account. If the intruder successfully guesses the password in the t^{th} attempt, then it will take all of A's money. The intruder does not have any incentive to continue its attack because the account balance is zero and there is a cost of launching an attack. The second condition of the above strategy also holds intuitively because the chance of guessing the password correctly increases in subsequent runs, and so does the intruder's expected payoff. But we will later show that the latter can be formally derived from the result of Theorem 2.

Note that as a part of an attack on a password-based protocol the intruder will interact with honest nodes to check the accuracy of its guess of the password. Obviously the intruder can manipulate protocol messages without guessing the password, but this does not improve its chance in subsequent attempts and hence is not optimal. Also there is no harm in modifying exchanged data and guessing the password at the same time. The readers might question what if the intruder blocks communication of a protocol run, but then it will get nothing, i.e. neither the prospect of a successful attack nor the benefit from additional behaviours of a generous initiator.

We summarise the intruder's cumulative payoff and probability that it is successful or unsuccessful up to k attempts in Table 3.

The following theorem shows that as k increases the probability α that the initiator A pursues additional behaviours upon a successful protocol session goes up but very slowly.

Theorem 2. Suppose that an intruder is allowed to attack a password-based protocol up to k runs for any $k \in$

[3]Our work only applies to online use of passwords, which is very different from offline setting, where Boyen [2] previously used a game-theoretic approach to quantify the security of his proposed protocol for offline use of password.

[4]This is true when the password is unchanged throughout a multiple-run attack.

No. of attempts	Outcome	Probability	Payoff of intruder
1	Succeed	$\epsilon = \epsilon_1 = 1/n$	U^+
2	Succeed	$(1-\epsilon_1)\epsilon_2 = 1/n$	$U^- + U^+$
3	Succeed	$(1-\epsilon_1)(1-\epsilon_2)\epsilon_3 = 1/n$	$2U^- + U^+$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
t	Succeed	$\epsilon_t \Pi_{i=1}^{t-1}(1-\epsilon_i) = 1/n$	$(t-1)U^- + U^+$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
k	Succeed	$\epsilon_k \Pi_{i=1}^{k-1}(1-\epsilon_i) = 1/n$	$(k-1)U^- + U^+$
k	Fail	$\Pi_{i=1}^{k}(1-\epsilon_i) = (n-k)/n$	kU^-

Table 3: This tables shows the cumulative payoff and probability of the intruder's success and failure when (s)he attacks a password-based protocol up to k runs.

$\{1, \ldots, n = 1/\epsilon\}$, and the intruder quits iff (s)he is successful in the t^{th} attempt where $t \leq k$ or fails in all k attempts as seen in Table 3. Then to discourage the intruder from attacking protocol runs between honest nodes, this inequality must hold:

$$\alpha > \frac{\epsilon U^+ + (1-\epsilon)U^-}{U} + \left(\frac{U^+ - U^-}{U}\right)\frac{k-1}{n(2n-k+1)}$$

PROOF. When an intruder attacks a protocol up to k runs, from Table 3, the expected (average) number of protocol runs the intruder intervenes is

$$N = \frac{1}{n} + \frac{2}{n} + \frac{3}{n} + \cdots + \frac{k}{n} + \frac{k(n-k)}{n} = \frac{k(2n-k+1)}{2n}$$

Similarly, the expected cumulative payoff of the intruder's multiple-run attack can be computed as follows

$$\begin{aligned} P &= \frac{U^+}{n} + \frac{U^- + U^+}{n} + \cdots + \\ &\quad \frac{(k-1)U^- + U^+}{n} + \frac{k(n-k)U^-}{n} \\ &= \frac{kU^+}{n} + U^-\left[\frac{1}{n} + \frac{2}{n} + \cdots + \frac{k-1}{n} + \frac{k(n-k)}{n}\right] \\ &= \frac{kU^+}{n} + \frac{k(2n-k-1)U^-}{2n} \end{aligned}$$

Since the expected payoff an intruder gets from not attacking a protocol in each run is αU, in order to discourage the intruder from attacking a password-based protocol up to k runs, we must have $\alpha U N > P$ or

$$\begin{aligned} \alpha &> \frac{kU^+}{nUN} + \frac{k(2n-k-1)U^-}{2nUN} \\ \alpha &> \frac{2U^+}{(2n-k+1)U} + \frac{(2n-k-1)U^-}{(2n-k+1)U} \\ \alpha &> \left(\frac{1}{n} + \frac{k-1}{n(2n-k+1)}\right)\frac{U^+}{U} + \\ &\quad \left(1 - \frac{1}{n} - \frac{k-1}{n(2n-k+1)}\right)\frac{U^-}{U} \\ \alpha &> (\epsilon + \Delta)\frac{U^+}{U} + (1 - \epsilon - \Delta)\frac{U^-}{U} \\ \alpha &> \frac{\epsilon U^+ + (1-\epsilon)U^-}{U} + \left(\frac{U^+ - U^-}{U}\right)\Delta \end{aligned}$$

where $\Delta = \frac{k-1}{n(2n-k+1)}$. □

Since $n \geq k \geq 1$, as k increases so do Δ and α. This implies that when the intruder attacks more protocol runs then its expected payoff per attack also increases, which justifies the second condition of the optimal strategy for an attacking intruder mentioned earlier. We further have:

- Since $\epsilon > \Delta \geq 0$ the difference between the bounds for α with respect to single-run (see Theorem 1) and n-run attacks is $\Delta(U^+ - U^-)/U < \epsilon(U^+ - U^-)/U$, which can be very small given that the password is of a reasonable length. This therefore affects the current limit for the number of times of entering wrong passwords consecutively as usually set in banking applications. We will discuss this further in Section 5.
- If this protocol transformation can discourage a k-run attack, then it can also discourage a t-run attack for any $t \leq k$.

4. MULTIPLE-RUN ATTACK ON MANUAL AUTHENTICATION PROTOCOL

In contrast to password-based schemes, the chance of a successful attack ϵ on a manual authentication protocol run remains unchanged regardless of how many times an attack is launched. This property applies to all secure protocols of this type, e.g. oneway, pairwise or group authentication [7, 11, 12, 15, 16, 17, 10, 23].

Our analysis here applies to many manual authentication protocols, but for clarity we give a pairwise protocol of the author [15]. In this scheme, parties A and B want to authenticate their public data $m_{A/B}$ from human interactions to remove the need of passwords, private keys and PKIs. The single arrow ($\longrightarrow$) indicates an unreliable and high-bandwidth link (e.g. WiFi or the Internet), whereas the double arrow ($\Longrightarrow$) represents an authentic and unspoofable channel. The latter is not a private channel (i.e. anyone can overhear it) and it is usually very low-bandwidth since it is implemented by humans, e.g., human conversations, text messages or manual data transfers between devices. $hash()$ and $uhash()$ are cryptographic and universal hash functions. Long random keys $k_{A/B}$ are generated by A/B, and k_A must be kept secret until after k_B is revealed in Message 2. Operators $\|$ and $\oplus$ denote concatenation and exclusive-or.

No. of attempts	Outcome	Probability	Payoff of intruder
1	Succeed	ϵ	U^+
2	Succeed	$\epsilon(1-\epsilon)$	$U^- + U^+$
3	Succeed	$\epsilon(1-\epsilon)^2$	$2U^- + U^+$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
t	Succeed	$\epsilon(1-\epsilon)^{t-1}$	$(t-1)U^- + U^+$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
k	Succeed	$\epsilon(1-\epsilon)^{k-1}$	$(k-1)U^- + U^+$
k	Fail	$(1-\epsilon)^k$	kU^-

Table 4: This tables shows the cumulative payoff and probability of the intruder's success and failure when (s)he attacks a manual authentication protocol up to k runs.

A pairwise manual authentication protocol [15]		
1. $A \longrightarrow B$	:	$m_A, hash(k_A)$
2. $B \longrightarrow A$	:	m_B, k_B
3. $A \longrightarrow B$	:	k_A
4. $A \Longleftrightarrow B$	:	$uhash(k_A \oplus k_B, m_A \parallel m_B)$

To ensure both parties share the same data, the human owners of devices A and B have to compare a short universal hash value (or a short authentication string SAS) of 16–32 bits manually. Since the universal hash key $k_A \oplus k_B$ always varies randomly and uniformly from one to another protocol run, the chance of a successful attack on each protocol run ϵ equals the collision probability of the universal hash function.[5]

Definition 1. [19] An ϵ-almost universal hash function, $uhash : R \times X \to Y$, must satisfy that for every $m, m' \in X$ and $m \neq m'$:

$$\Pr_{\{k \in R\}}[uhash(k, m) = uhash(k, m')] \leq \epsilon$$

As discussed in [17, 16], for a b-bit universal hash function the best-possible ϵ is 2^{-b} and there are various constructions that achieve close to this.

To discourage the intruder from attacking a manual authentication protocol in multiple runs, we use the protocol transformation of Table 1. Upon a successful protocol session, with probability α the initiator A pursues additional behaviours that benefit the intruder.

Since the chance of a successful single-run attack ϵ is unchanged, intuitively the value of α required to discourage a multiple-run attack is the same as in a single-run attack of Theorem 1. But we will formally state and prove this result in Theorem 3.

Theorem 3. Suppose that an intruder is allowed to attack a manual authentication protocol up to k runs for any $k \geq 1$, and the intruder quits iff (s)he is successful in the t^{th} attempt where $t \leq k$ or fails in all k attempts as seen in Table 4. Then to discourage the intruder from attacking protocol runs between honest nodes, this inequality must hold:

$$\alpha > \frac{\epsilon U^+ + (1-\epsilon)U^-}{U}$$

We summarise the intruder's cumulative payoff and probability of success and failure in Table 4.

PROOF. When an intruder attacks a protocol up to k runs, from Table 4, the expected number of runs the intruder intervenes in this protocol is:

$$\begin{aligned} N &= \epsilon + 2\epsilon(1-\epsilon) + \cdots + k\epsilon(1-\epsilon)^{k-1} + k(1-\epsilon)^k \\ &= 1 + (1-\epsilon) + (1-\epsilon)^2 + \cdots + (1-\epsilon)^{k-1} \qquad (1) \end{aligned}$$

Equality (1) is derived from repeatedly applying this equality for all $t \in \{1, \cdots, k-1\}$:

$$(1-\epsilon)^t = t(1-\epsilon)^{t-1}\epsilon + (t+1)(1-\epsilon)^t - t(1-\epsilon)^{t-1}$$

The expected cumulative payoff of the intruder's multi-run attack is:

$$\begin{aligned} P &= \epsilon U^+ + \epsilon(1-\epsilon)(U^- + U^+) + \cdots + \\ &\quad \epsilon(1-\epsilon)^{k-1}((k-1)U^- + U^+) + (1-\epsilon)^k kU^- \\ &= U^+\epsilon\left[1 + (1-\epsilon) + \cdots + (1-\epsilon)^{k-1}\right] + \\ &\quad U^-(1-\epsilon)\left[\epsilon + \cdots + (k-1)\epsilon(1-\epsilon)^{k-2} + k(1-\epsilon)^{k-1}\right] \\ &= U^+\epsilon N + U^-(1-\epsilon)N = N\left[U^+\epsilon + U^-(1-\epsilon)\right] \end{aligned}$$

Since the expected payoff an intruder gets from not attacking each protocol run is αU, to discourage the intruder from attacking a protocol in multiple runs, we must have $\alpha UN > P$ or $\alpha > \frac{\epsilon U^+ + (1-\epsilon)U^-}{U}$ □

5. CASE STUDY

Let us suppose that A has perhaps accidentally revealed his financial details, e.g., bank statement to someone whom A later distrusts. A then wants to discourage that person or the attacker from interfering with online transactions carried out between him and the bank because (1) A can have a large amount of money in his account and (2) A wants to have the freedom to carry out transactions with other parties without being disrupted by the attacker. In the following scenario, the attacker plays the role of a lending company, but our work is applicable to other situations where the above condition applies.

- Party A has borrowed some money from a lending company.
- A however delays making the payment because of, e.g., further investment, and this goes against the interest of the lender who does not trust A.
- To make a loan, the lender must have seen A's financial proofs such as bank statements, and therefore knows that A potentially has a bank account of up to 300 thousand US dollars. With this information the lender will be extremely tempted to break into A's account and get all of A's money.

Whenever A carries out an online transaction, he authenticates himself to the bank by typing in his 20-bit password

[5] We note that our protocol transformation of Table 1 and the analysis of this section also apply to other manual authentication protocols, including schemes of Vaudenay [22], Čagalj et al. [4], and Hoepman [9, 15] which do not use a universal hash function.

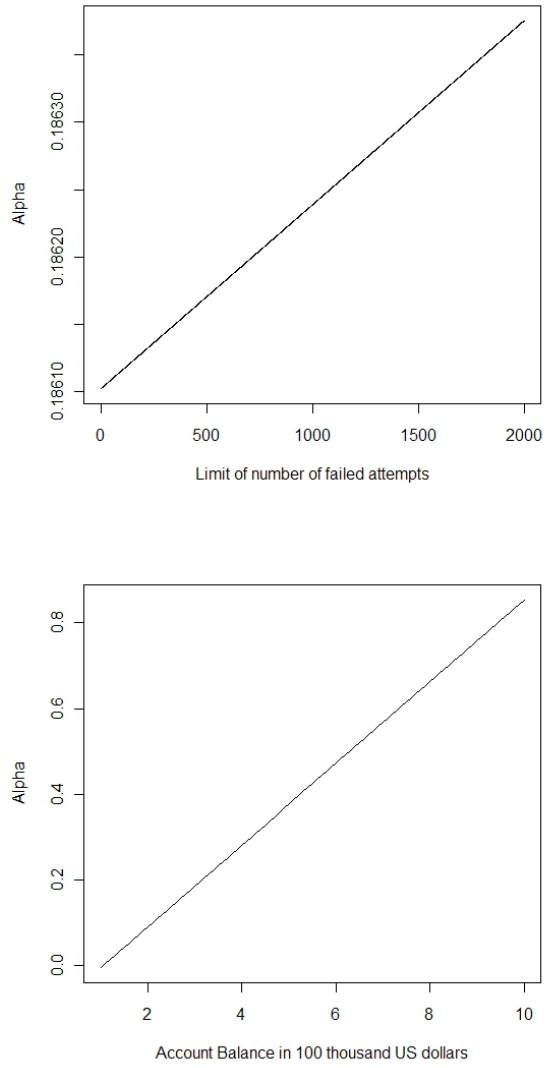

Figure 1: Experimental results.

(or a 5-hexadecimal-digit number) on the bank's website, and hence $\epsilon = 2^{-20}$ or $n = 2^{20}$.

Let us suppose that it costs the lender 0.1 US dollar[6] to launch an attack on the authentication stage of the online banking protocol then upon a successful attack the payoff for the lender is $U^+ =$ (300,000 - 0.1) US dollars. If the lender fails, it gets a negative payoff $U^- = -0.1$ US dollar due to the running cost.

We therefore have this inequality $\epsilon U^+ + (1-\epsilon)U^- > 0$, which implies that there is no harm for the lender to attack the authentication stage of online transactions. To discourage the lender from misbehaving, A will pursue the following additional behaviour. Each time after A authenticates himself to the bank successfully, with probability $\alpha \in [0, 1)$ the account holder or borrower will make a small payment of 1 US dollar to the lender. The payoff for the lender after a successful protocol run which it does not interfere is therefore $U = 1$ US dollar, i.e. of course the lender does not get this dollar when it decides to attack an online transaction but fails.

It might be worth to point out that it is not frequent that account holders accidentally reveal personal financial data to untrustworthy parties, and hence the number of potential intruders attacking a single account holder should be small and the small payment suggested above would be practical.

In the first graph of Figure 1, we use results from Theorem 2 to plot α against $k \in \{1, \cdots, 2000\}$ where k is the maximum number of attacks the lender can launch. In this experiment ϵ, n, U, U^+, U^- are fixed as defined earlier.

$$\alpha = \frac{\epsilon U^+ + (1-\epsilon)U^-}{U} + \left(\frac{U^+ - U^-}{U}\right)\frac{k-1}{n(2n-k+1)}$$

It is very clear that as k increases so does α but very slowly, i.e. α is around 0.186 for any $k \in \{1, \cdots, 2000\}$. We argue that this can have a significant impact on many banking applications which usually set $k = 3$ and so can be inconvenient to use especially after one comes back from a holiday and there are too many passwords to remember. What this shows is that the number of consecutive attempts of entering a wrong password can be increased significantly without compromising the economic security of online banking protocols.

Since the account balance and hence U^+ can also vary, in the second graph of Figure 1 we use results from Theorem 2 to plot α against U^+ ranging from 100 to 1000 thousand US dollars, and k is set at 1000. It is clear that as U^+ increases so does α. To keep α small, one can increase the password bitlength to reduce the likelihood ϵ of a successful attack, but longer passwords are harder to remember and require more human interactions. Also the password length is usually fixed for each type of bank account regardless of how much money a customer puts in. What this shows is that it is wise to leave only a large enough amount of money in each account to keep α low.

The readers might question what if A does not pay the lender the small amount of money, even though A had agreed to it. The answer is as follows: the lender will regularly monitor its own account to check whether this small payment occurs with probability α with respect to the total number of successful transactions carried out by A. If this agreement were violated, the lender would change its mind and re-launch its attack immediately. However what we also need to keep in mind is a trade-off between U and α. Given the parameters ϵ, U^+, U^-, k are fixed, then αU is also fixed according to the above formula. Consequently as α increases U decreases. Obviously A does not want to waste too much money on transaction cost, and hence we would prefer to keep α small. This is very similar to the idea of micropayment first introduced by Rivest [18].

We have illustrated the use of the protocol transformation in banking, we however have to regconise that our model as introduced so far has limitations. In particular, we have only considered banking applications where the identity of the intruder is known to or suspected by the protocol initiator or participants, even though such a restriction reduces the number of potential attackers and hence makes the protocol transformation feasible. In addition, knowledge of at-

[6] According to `www.csgnetwork.com/elecenergycalcs.html` the cost of running a home computer system for one hour is 0.08 US dollar.

tacker's payoffs with respect to different protocol outcomes is required to be known to the initiator in advance. Since authentication is also used in e-mail communication, web page and many others, it would be interesting to investigate how one can quantify the gain for the attacker in the applications.[7]

6. EXTENDED PROTOCOL TRANSFORMATION

We have so far focused on the use of the protocol transformation of Table 1 on pairwise authentication protocols where there is a designated role for the protocol initiator who is generous with probability α after each successful protocol session. Although the same protocol transformation is applicable to group authentication schemes, it might be difficult for multiple protocol participants to agree on who will be the initiator. We therefore would like to remove the need for such an initiator by assuming that every honest node can be generous with probability α independently. In other words, we extend our protocol transformation and apply it to the behaviour of every group protocol participant, and hence the *extended protocol transformation*.

We will first address the payoff assignment issue before going into details of how the extended protocol transformation benefits both password-based and manual authentication protocols in Subsections 6.1 and 6.2 respectively.

In a group authentication protocol, there are always two or more nodes in a group $\mathbf{G}$ who want to authenticate or agree on the same data. For the types of considered protocols not all of the protocol participants have to be honest, and this means that these compromised principles are not obliged to follow our protocol transformation. We will discuss further what the compromised nodes might do later because their behaviours are dependent on the type of authentication protocols and hence different attacking models, such as the *Machiavelli* adversary model [21], are needed. We denote p the number of honest parties out of all protocol participants and $p \geq 1$.

When the intruder does not attack a protocol run, there are two main possibilities that affect the payoff of the intruder:

- With probability $(1-\alpha)^p$, every honest node is ungenerous and there is no payoff for the intruder.
- With probability $1-(1-\alpha)^p$, there is at least one generous node. The payoff for the intruder might vary according to the number of generous protocol participants, but as in pairwise schemes we only consider the worst-case scenario where the intruder's payoffs is always the same under this condition.

When the intruder attacks a protocol run, there are also two possibilities that affect the intruder's payoff:

[7] As regards e-mail communication, one can treat attacks on the application as the first step to achieve a bigger goal, e.g., breaking a bank account. For example, through illegal access to one's e-mail account the intruder might obtain personal and confidential data including bank account details but not the secret password. Using this information, the intruder will be motivated to break into the e-mail account holder's bank account subsequently. The same observation is applicable to web page applications which also potentially possess private information of online customers.

Strategy of intruder	Outcome of protocol	Strategies of honest nodes	Payoff of intruder
No attack	Succeed	All ungenerous	0
No attack	Succeed	≥ 1 generous node	U
Attack	Succeed	Any	U^+
Attack	Fail	All ungenerous	U^-

Table 5: A summary of the game.

- With probability ϵ, the attack is successful. The payoff for the intruder also can vary according to the number of generous protocol participants, but again we only consider the worst-case scenario where the intruder's payoff is the same here.
- With probability $1-\epsilon$, the intruder fails and gets a negative payoff.

We summarise the payoffs for the intruder in different scenarios in Table 5.

Based on the damages an intruder might cause to honest parties, it is clear from Table 5 that we always have:

$$U^+ > U > 0 > U^-$$

6.1 Password-based group authentication protocols

For clarity, we present the group version of the pairwise Diffie-Hellman-based Encrypted Key Exchange scheme of Bellovin and Merritt [1]. This protocol establishes a shared private key $g^{x_A x_B}$ between any two parties A and B, where g^{x_A} and g^{x_B} are public Diffie-Hellman keys of A and B, from a short password pw using an encryption scheme $E_{pw}()$ and a cryptographic hash function $hash()$.

Group password-based Encrypted Key Exchange Protocol
1. $\forall A \longrightarrow \forall B \ : \ A \parallel E_{pw}(g^{x_A})$
2. $\forall B \longrightarrow \forall A \ : \ hash(sk_{AB} \parallel 1)$
and $sk_{AB} = hash(A \parallel B \parallel g^{x_A} \parallel g^{x_B} \parallel g^{x_A x_B})$
3. $\forall A \longrightarrow \forall B \ : \ hash(sk_{AB} \parallel 2)$

In a password-based group protocol, such as the one above, all parties in group $\mathbf{G}$ share a common and private password pw. Following the *Machiavelli* adversary model introduced by Syverson et al. [21], we will allow the presence of compromised protocol participants. But, to make these protocols usable, those compromised principles are restricted to the following behaviours:

- In an attempt to collaborate with the intruder, compromised protocol participants will not obey our extended protocol transformation.
- However compromised principles will not share the password with the intruder or anyone else outside group $\mathbf{G}$.
- Additionally compromised nodes will not use their knowledge of the shared password to fool other honest protocol participants into agreeing on the same and corrupt keys.

The latter two conditions must hold, for otherwise it is impossible to resist an intruder who possesses the password. Thus the intruder will not receive much support from compromised or Machiavellian parties.

Using information from Tables 3 and 5, we arrive at the following theorem whose proof is very similar to the proof of Theorem 2, and hence is not provided.

Theorem 4. Suppose that an intruder is allowed to attack a group password-based protocol up to k runs for any $k \geq 1$. There are p honest protocol participants, and any of whom is generous with probability α independently. If the following inequality holds then the intruder will not have any incentive to attack protocol runs of honest nodes.

$$1-(1-\alpha)^p \quad > \quad \frac{\epsilon U^+ + (1-\epsilon)U^-}{U} + \left(\frac{U^+ - U^-}{U}\right)\frac{k-1}{n(2n-k+1)}$$

We note that there is no change in the right hand side of the above condition relative to Theorem 2, because the intruder's payoffs in a group protocol from Table 5 are very similar to pairwise schemes of Table 2. The left hand side has however become $1-(1-\alpha)^p$ which is equivalent to α when $p = 1$ or there is one honest node following the protocol transformation.

The same observation is applicable to group manual authentication protocols provided in the next section.

6.2 Group manual authentication protocols

For clarity, we give the specification of a group manual authentication protocol which is similar to group manual authentication schemes of the author [16, 15]. In this scheme, all parties As of group $\mathbf{G}$ want to authenticate their public data m_A's from human interactions to remove the need of passwords, private keys and PKIs. Each long random key k_A generated by $A \in \mathbf{G}$ must be kept secret until after A has received Messages 1 from all other party $B \in \mathbf{G}$.

Manual group authentication protocol [16]
1. $\forall A \longrightarrow \forall B : m_A, hash(A, k_A)$
2. $\forall A \longrightarrow \forall B : k_A$
3. $\forall A \Longrightarrow \forall B : uhash(k^*, M)$
where k^* is the XOR of all k_A's for $A \in \mathbf{G}$
and M is the concatenation of all m_A's for $A \in \mathbf{G}$

Using information from Tables 4 and 5, we arrive at the following theorem whose proof is very similar to the proof of Theorem 3 and hence is omitted.

Theorem 5. Suppose that an intruder is allowed to attack a group manual authentication protocol up to k runs for any $k \geq 1$. Moreover there are p honest protocol participants, and any of whom is generous with probability α independently. If the following inequality holds then the intruder will not have any incentive to attack protocol runs of honest nodes.

$$1-(1-\alpha)^p > \frac{\epsilon U^+ + (1-\epsilon)U^-}{U}$$

7. CONCLUSIONS AND FUTURE RESEARCH

We have used ideas from game theory to transform two families of authentication protocols, namely password-based authentication and manual authentication protocols, to make them resilient against a rational intruder. In these protocols, only the intruder and dishonest parties are self-interested and all other trustworthy protocol participants should cooperate to complete a protocol run successfully, since it is in their mutual interest to agree on the same data.

At the heart of our protocol transformation is the introduction of some additional behaviours protocol participants can pursue to discourage the intruder. Although our work and in particular the additional behaviours might appear strange at first sight, we later discover that similar strategies have been exploited in not only rational secret sharing schemes [6, 8] but also the business of insurance and lottery.

As an example taken from insurance business, we usually pay a fixed amount of money regularly to an insurance company, because we will be compensated by the insurance company upon some bad events happening in our lives such as accidents. However, the premium we pay for insurance is substantially greater than the average cost of claims. But what persuade us to pay insurance are

- We want to be protected from accidents, even though they are as unlikely as the chance of a successful attack and
- The fixed amount of insurance fee paid monthly has a small impact on our lifestyle, as best illustrated by the convex utility function in economics.

The same observation, perhaps surprisingly, can be made to the additional behaviour that benefits the intruder in our protocol transformation. Namely the agreement to make a small payment with probability α introduced in our case study:

- gives honest parties the protection that their legitimate online financial transactions will not be disrupted or attacked by the intruder.
- does not have a major financial impact on the parties.

The main deficiency of the approach set out in this paper is that it does not provide a method of specifying the additional behaviour for each different application and the intruder's interest. Giving away a tiny amount of money might be suitable and practical for banking applications, but for other applications we need to come up with something else. We however note that our approach has already been applied to improve the security of distance-bounding protocols [15], where the choice of the threshold on the number of erroneous responses transmitted over a noisy communication in a rapid-bit exchange represents the additional behaviour that is controlled by the tag reader or the protocol initiator. This shows that giving away a tiny amount of money or exchanging useless data as suggested in the preliminary version of this work [14] are not the only behaviours that can be incorporated into our protocol transformation.[8]

9[8]The approach taken in this paper is that the protocol initiator will first wait until the end of each protocol run, and then decide his or her strategy. In the preliminary version of this work [14], we took a sightly different approach in the sense that the protocol initiator decides the strategy prior to each protocol run. Hence in an

While we have explored the notion of rational intruder in two types of authentication protocols, our work reported here opens the way to a number of new problems. For example, it would be interesting to investigate how relevant the notion of a rational intruder is to other types of authentication protocols which are based on PKIs or long private keys. Since our protocol transformation works best when protocols are immune to (off-line) searching, can it be relaxed or modified to accommodate a wider variety of possible attacks, e.g. substitution attacks that are relevant to other cryptographic primitives (including MACs)? One might also consider increasing rationality assumptions for the intruders, as in the *Macchiavellian* adversary [21] who does not share private keys and passwords with its collaborators.

Also our studies on password-based protocols hopefully would lead to further attempts in improving the usability and economic security of many banking applications based on passwords which are currently quite inconvenient to use regarding the limit on the number of consecutive failed attempts of entering the correct password.

8. REFERENCES

[1] S.M. Bellovin and M. Merritt. *Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks*. Proceedings of the IEEE Symposium on Research in Security and Privacy (Oakland): 72.

[2] X. Boyen. *Halting Password Puzzles – Hard-to-break Encryption from Human-memorable Keys*. USENIX Security Symposium - SECURITY '07, 2007.

[3] L. Buttyán, Jean-Pierre Hubaux, and S. Čapkun. *A Formal Analysis of Syverson's Rational Exchange Protocol*. Proceedings of the 15th IEEE CSF 2002.

[4] M. Čagalj, S. Čapkun and J. Hubaux, Key agreement in peer-to-peer wireless networks, in: *Proceedings of the IEEE Special Issue on Security and Cryptography* **94**(2) (2006), A. Mazzeo, ed., 467-478.

[5] C. Dimitrakakis, A. Mitrokotsa and S. Vaudenay. *Expected loss bounds for authentication in constrained channels*. In INFOCOM 2012 or `http://arxiv.org/pdf/1009.0278`

[6] G. Fuchsbauer, J. Katz and D. Naccache. *Efficient Rational Secret Sharing in Standard Communication Networks*. TCC 2010: 419-436

[7] C. Gehrmann, C. Mitchell and K. Nyberg. *Manual Authentication for Wireless Devices*. RSA Cryptobytes, vol. 7, no. 1, pp. 29-37, 2004.

[8] S.D. Gordon and J. Katz. *Rational secret sharing, revisited*. In Proceedings of Security and Cryptography for Networks. LNCS vol. 4116, 229-241, 2006.

[9] J.-H. Hoepman, *Ephemeral pairing problem/* Proceeding of the 8th International Conference on Financial Cryptography, LNCS, Vol. 3110, A. Juels, ed., Springer, 2004, pp. 212-226.

[10] ISO/IEC 9798-6 (revision), L.H. Nguyen, ed., 2010, *Information Technology – Security Techniques – Entity authentication – Part 6: Mechanisms using manual data transfer*.

authentication protocol where nodes exchange data authentically, it makes sense for the initiator to authenticate useless data occasionally as the additional behaviour to discourage the intruder to attack.

[11] Sven Laur and Kaisa Nyberg. *Efficient Mutual Data Authentication Using Manually Authenticated Strings*. LNCS Vol. 4301, pages 90-107, 2006.

[12] S. Laur and S. Pasini. *SAS-Based Group Authentication and Key Agreement Protocols*. In Public Key Cryptography - PKC 2008, 11th International Workshop on Practice and Theory in Public-Key Cryptography, pp. 197-213.

[13] L.H. Nguyen. *Rational distance-bounding protocols over noisy channels*. In the Proceedings of the 4th International Conference on Security of Information and Networks SIN 2011, pp. 49-56.

[14] L.H. Nguyen. *Rational authentication protocols*. The preliminary draft of this paper was posted in the IACR Cryptology ePrint Archive at `http://eprint.iacr.org/2011/070`.

[15] L.H. Nguyen and A.W. Roscoe. *Authentication protocols based on low-bandwidth unspoofable channels: A comparative survey*. Journal of Computer Security 19(1): 139-201 (2011).

[16] L.H. Nguyen and A.W. Roscoe. *Authenticating ad-hoc networks by comparison of short digests*. Information and Computation **206**(2-4) (2008), 250-271.

[17] L.H. Nguyen and A.W. Roscoe. *Short-output universal hash functions, and their use in fast and secure message authentication*. In the Proceeding of the 19th International Workshop on Fast Software Encryption FSE 2012.

[18] R.L. Rivest. *Peppercoin Micropayments*. Proceedings Financial Cryptography '04. (ed. Ari Juels) LNCS, Vol. 3110. (Springer, 2004), 2–8.

[19] D.R. Stinson. *Universal Hashing and Authentication Codes*. Advances in Cryptology - Crypto 1991, LNCS vol. 576, pp. 74-85, 1992.

[20] P. Syverson. *Weakly secret bit commitment: Applications to lotteries and fair exchange*. The IEEE Computer Security Foundations Workshop, pages 2-13, 1998.

[21] P. Syverson, C. Meadows, I. Cervesato. *Dolev-Yao is no better than Machiavelli*. First Workshop on Issues in the Theory of Security 2000.

[22] S. Vaudenay. *Secure Communications over Insecure Channels Based on Short Authenticated Strings*. Crypto 2005, LNCS vol. 3621, pp. 309-326.

[23] J. Valkonen, N. Asokan and K. Nyberg. *Ad Hoc Security Associations for Groups*. In Proceedings of the Third European Workshop on Security and Privacy in Ad hoc and Sensor Networks 2006. LNCS vol. 4357, pp. 150-164.