Karlsruhe Institute of Technology

Complete Primitives for Information-Theoretically Secure Two-Party Computation

zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

von der Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Daniel Kraschewski

aus Bad Pyrmont

Tag der mündlichen Prüfung:	25. Januar 2013
Erster Gutachter:	Prof. Dr. Jörn Müller-Quade
Zweiter Gutachter:	Prof. Dr. Yuval Ishai

Acknowledgements

First of all, I want to express my deep gratitude to Jörn Müller-Quade. I am very grateful for his support and the exceptional freedom of research I enjoyed under his doctorate supervision. His fascination for the wonders of modern cryptography were a great source of inspiration, and I personally benefited also a lot from numerous non-technical discussions about science in general, ways of life and philosophy. No less I want to thank Yuval Ishai for co-refereeing this thesis and all his support far beyond that. I am especially impressed, how every discussion with him was characterized by a notably kind atmosphere and always turned out fruitful.

For proofreading, help with LaTeX on various occasions and his invaluable friendship I am in great debt to my former colleague Christian Henrich. I am also very thankful to all my colleagues for the familiar working atmosphere. In particular, I want to thank Nico Döttling for always having time for any technical discussion, and I thank Dennis Hofheinz for many helpful comments and the honor and pleasure to work with him. My sincerest thanks go to Carmen Manietta and Holger Hellmuth, who at all times wondrously managed to keep any administrative and technical trouble behind the scenes.

Special thanks go to the mathoverflow community and especially Felipe Voloch for pointing me to the Łojasiewicz Inequality.

Last but not least, I would like to express my heartfelt gratitude to my family members and personal friends who encouraged and supported me throughout my studies.

Contents

Abstract

In 1988, Joe Kilian showed that arbitrary multi-party computation can be securely realized from a quite simple primitive, namely *oblivious transfer* (OT). This primitive in its basic form is just a trusted erasure channel: The sender can enter a bit of his choice, which is then transferred to the receiver with probability $\frac{1}{2}$ and otherwise replaced by a special erasure symbol. Since the discovery that OT is complete in the above-mentioned sense, cryptographers are working on reductions of OT (and thus general secure multi-party computation) to various other primitives. The present thesis contributes two results to this research area.

The first contribution of this thesis exhaustively solves the long-standing open question, which *cryptogates* allow for information-theoretically secure implementation of OT and are thus complete, too. I.e., comprehensive but easily checkable completeness criteria are provided for any trusted black box that can be jointly queried by two parties, has finite input and output alphabets, and does not change behavior depending on time or input history. The criteria existing so far only cover special classes of cryptogates, e.g. cryptogates that do not use any internal randomness, or noisy channels (i.e., one party gets no output and the other party cannot provide any input). The novel approach of this thesis, by which the limitations of former results are overcome, is the definition and thorough investigation of a very specific algebraic structure of "idealized cheating strategies". Then, powerful estimation techniques from probability theory and real algebraic geometry are adapted to base the cryptographic security proof for a generic reduction protocol on the mathematical properties of this structure.

The second contribution of this thesis builds on a tamper-proof hardware assumption, where the hardware issuer is one of the mutually mistrusting parties. In the literature one finds a rash of protocol constructions for OT based on untrusted tamper-proof hardware, aiming at

- a decrease of the number of exchanged hardware tokens,
- saving communication and computation costs, and
- reduction of the required computational assumptions.

The approach in this thesis needs only a single token to be exchanged and has asymptotically optimal communication overhead. The computational costs are still remarkably low, and there just are not any computational assumptions used. This comes at the cost of only bounded token reusability and a fairly involved security proof. However, unbounded token reusability can be achieved straightforwardly by the weakest common complexity assumption, namely the existence of a pseudorandom number generator. The basis for these results is a special technique for implementation of affine functions on a tamper-proof token, such that the token receiver can verify correctness of the implementation but does not learn the concrete function parameters.

Zusammenfassung

Moderne Kryptographie ist weit mehr als die Wissenschaft der Verschlüsselung. Datenschutzkonformer Abgleich von Fahndungslisten, elektronische Wahlverfahren, e-Commerce und betriebswirtschaftliches Benchmarking stellen Herausforderungen, welche allein mit abhör- und manipulationssicherem Datentransfer nicht gelöst werden können. Für den Entwurf entsprechender Protokolle werden hinreichend mächtige Primitive benötigt.

In meiner Dissertation liefere ich ein einfaches kombinatorisches Kriterium, mittels welchem für jede zustandslose Zweiparteien-Primitive mit endlichem Eingabealphabet (effizient) entscheidbar ist, ob sie für allgemeine sichere Berechnungen ausreicht. Ferner zeige ich, dass es für sichere Zweiparteien-Berechnungen bereits hinreichend ist, wenn eine der beteiligten Parteien ein manipulationssicheres Hardware-Token erstellen kann (welchem die andere Partei in keiner Weise vertraut).

Hintergrund: kryptographisch sichere Berechnungen

Der Ursprung des Forschungsgebiets der sog. „sicheren Mehrparteien-Berechnungen“ wird gemeinhin in einer 1982 von Andrew Yao aufgestellten Frage gesehen, welche als „Yaos Millionärsproblem“ bekannt geworden ist:

> Wie können zwei sich gegenseitig misstrauende Millionäre herausfinden, wer von ihnen reicher ist, ohne dass einer von ihnen irgendeine weitergehende Information über den konkreten Wert seines Besitzes offenlegen muss?

Die Verallgemeinerung dieses Problems ist recht naheliegend: Eine Gruppe von Parteien $P_1, \ldots, P_n$ will gemeinsam einen Funktionswert $f(x_1, \ldots, x_n)$ berechnen mit geheimer Eingabe x_i von Partei P_i. Keine Partei P_i, selbst wenn sie sich beliebig bösartig verhält, darf dabei etwas über $(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ erfahren, was nicht direkt aus ihrer Eingabe x_i und dem öffentlichen Funktionsergebnis $f(x_1, \ldots, x_n)$ berechnet werden kann. Außerdem soll die Berechnung sicher gegen Verfälschung des Ergebnisses sein, d. h. keine Partei P_i darf das Berechnungsergebnis anders beeinflussen können als durch entsprechende Wahl ihrer Eingabe x_i. Zudem sollen selbst gegenüber Gruppen aus mehreren bösartig kollaborierenden Parteien entsprechende Sicherheitseigenschaften gelten.

Hauptergebnisse der Dissertation

Vollständigkeitssatz für zustandslose Zweiparteien-Primitive. Joe Kilian konnte 1988 zeigen, dass jede beliebige Mehrparteien-Berechnung kryptographisch sicher auf einer Primitive namens „Oblivious Transfer“ (OT) aufbauend realisiert werden kann. Diese Primitive erlaubt es einer Partei, zwei Bits s_0, s_1 an einen dedizierten Empfänger zu senden, sodass der Empfänger nur eines der beiden lernt, im Folgenden s_c genannt. Der Empfänger kann die Auswahl c selbst festlegen, lernt aber nichts über s_{1-c}; umgekehrt bleibt c dem Sender gegenüber geheim. Das Ergebnis von Kilian wirft die natürliche Frage auf, welche anderen Primitive ebenfalls in diesem Sinne vollständig sind. Für zustandslose, deterministische Zweiparteien-Primitive mit endlichem Eingabealphabet und symmetrischer Ausgabe (beide Parteien erhalten dasselbe Ergebnis) wurde diese Frage 1991 und für Primitive mit asymmetrischer Ausgabe (nur eine der beiden Parteien erhält das Berechnungsergebnis) im Jahr 2000 von Kilian selbst beantwortet.

In meiner Dissertation werden die Vollständigkeitskriterien von Kilian vereinheitlicht und auf beliebige zustandslose (aber nicht mehr notwendigerweise deterministische) Zweiparteien-Primitive mit endlichem Eingabealphabet erweitert. Damit wird die Vollständigkeitsfrage erstmals auch für Zweiparteien-Primitive geklärt, welche unterschiedliche Ergebnisse an die beteiligten Parteien ausgeben und/oder in die Berechnung internen Zufall einfließen lassen. Hierzu ist anzumerken, dass sich die Ansätze von Kilian nicht ohne Weiteres verallgemeinern lassen, zumal er fundamental verschiedene Techniken für den symmetrischen und den asymmetrischen Fall verwendet. Meine neuartige Herangehensweise besteht in einer allgemeinen Protokollkonstruktion, für welche alle perfekt unentdeckbaren Angriffsstrategien gewissen Polynomgleichungen genügen müssen. Damit lässt sich die Menge aller perfekten Angriffe als entsprechende Nullstellenmenge (algebraische Varietät) beschreiben und ich kann Protokollparameter angeben, für welche ausschließlich triviale Angriffe existieren, die die Sicherheit nicht bedrohen. Unter Verwendung geeigneter Abschätzungsmethoden aus Wahrscheinlichkeitstheorie (Hoeffding-Ungleichung) und der reellen algebraischen Geometrie (Łojasiewicz-Ungleichung) kann ich außerdem zeigen, dass *jeder* Angriff hinreichend nahe an einer perfekten Angriffsstrategie liegt, sodass die Sicherheit meiner Protokollkonstruktion gegen perfekte Angriffe bereits Sicherheit gegen allgemeine Angriffe impliziert.

Sichere Mehrparteien-Berechnungen mittels manipulationssicherer Hardware. Für die Realisierung sicherer Mehrparteien-Berechnungen sind kryptographische Grundannahmen unabdingbar. Darüberhinaus sind besonders restriktive Sicherheitsbegriffe wie die sog. „universelle Komponierbarkeit“ allein mit Komplexitätsannahmen (z. B., dass die Faktorisierung großer Zahlen nicht praktikabel ist) beweisbar nicht zu erfüllen; es werden zusätzliche Setup-Annahmen benötigt (z. B., dass eine Public-Key-Infrastruktur gegeben ist). Manipulationssichere Hardware bietet hier einen alternativen Ansatz und überraschenderweise darf die Hardware sogar von einer der sich gegenseitig misstrauenden Protokollparteien stammen. In der Literatur sind entsprechende Konstruktionen für sichere Zweiparteien-Berechnungen zu finden, die die prinzipielle Machbarkeit demonstrieren.

In meiner Dissertation stelle ich das erste Resultat für informationstheoretisch sichere, universell komponierbare Zweiparteien-Berechnungen vor, welches lediglich den Austausch eines *einzigen* Hardware-Tokens benötigt. Frühere Konstruktionen benötigten entweder zusätzliche Komplexitätsannahmen oder es musste eine Vielzahl an Token ausgetauscht werden. Des Weiteren werden die bekannten Lösungen aus der Literatur dahingehend übertroffen, dass ich für verschiedene Primitive (darunter auch OT) erstmals informationstheoretisch sichere Protokolle mit optimaler Kommunikationskomplexität angeben kann. Auch der benötigte Rechenaufwand fällt auffallend gering aus, was insofern von spezieller Bedeutung ist, als manipulationssichere Hardware-Token i. A. nicht als leistungsstark angenommen werden können. Basis für diese Resultate ist eine neu entwickelte Technik, mit der sich auf besonders effiziente Weise affine Funktionen über endlichen Körpern so auf dem Token implementieren lassen, dass gegenüber einer misstrauischen Empfängerpartei ohne Offenlegung der Funktionsparameter die Korrektheit nachgewiesen werden kann.

Preamble

Background and motivation

Secure multi-party computation/secure function evaluation. Modern cryptography is far more than development and analysis of cypher schemes. Today, there are many challenges that cannot be dealt with just by secure message transfer:

- comparing national wanted lists without violating data privacy laws,
- benchmarking of business competitors that refuse to disclose their business data,
- electronic elections,
- online poker without a trusted game server,

to name only a few. Basically, all these problems necessitate some kind of "game rules" (protocol) that allow the involved parties to commonly perform the desired computation without the need to trust each other. The design and security analysis of such protocols is subject of the research area of *secure multi-party computation* (MPC), whose origin goes back to Yao's Millionaire's Problem:

> "Two millionaires wish to know who is richer; however, they do not want to find out inadvertently any additional information about each other's wealth. How can they carry out such a conversation?" [Yao82]

A bit more formally, the fundamental task of MPC consists in the following, quite natural generalization of Yao's Millionaire's Problem: Some parties $P_1, \ldots, P_n$ want to commonly evaluate a function $f(x_1, \ldots, x_n)$, where x_i is a secret input from party P_i, and no party P_i should learn more about the other parties' secrets $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$ than what can be inferred from its own input x_i and the public computation result $f(x_1, \ldots, x_n)$. Further, no party P_i should be able to influence the final computation result $f(x_1, \ldots, x_n)$ other than by choosing x_i. This task is usually referred to as *secure function evaluation* (SFE).

Given a general SFE solution, one could cope with all the challenges presented at the beginning, some of which can be reformulated as an SFE instance more obviously than others. In particular, the reduction to SFE is fairly straightforward for privacy preserving comparison of wanted lists, business benchmarking and electronic elections.

comparison of wanted lists: The parties' secret inputs x_i are sets of data records (one record for each wanted person), and the function f computes and outputs a simple set intersection. More sophisticated variants, where similar but not perfectly matching records are also part of the output, are as well possible.

benchmarking: This is the most straightforward example. Each party inputs its private business data, and the function output is the respective benchmark.

electronic elections: At its basics, an electronic election can also be translated into the terms of SFE very simply; the votes are the secret inputs and the tally is the public function outcome. Complex elections might consist of more than one round, but still in most cases this can just be handled as a sequence of individual elections.

Online poker stands out from our list in the sense that the respective reduction to SFE requires some more sophisticated techniques, which was the main reason to include also such a not so serious example. First of all, a poker game needs some trusted source of randomness. Since no party is trusted by the others, we have no designated dealer to "shuffle" the cards. Secondly, a poker game consists of several rounds, which cannot be treated independently. Last but not least, some of the players' information during the game is non-public, since nobody can see the others' cards. Thus, the intermediate game state cannot be a public function output. However, all these issues can be solved by means of SFE.

Providing the function f in an SFE protocol with some additional randomness can be done by the following generic trick. If l random bits are needed, each participant just has to additionally input a uniformly random l-bit string. Then, if at least one party honestly follows the protocol, the bitwise XOR of these additional input strings can be used as trusted randomness. Note that it

usually suffices to protect only the honest parties, and thus there is no need to care about the output distribution if *all* parties are corrupted. Now, with such a randomized SFE solution one can already implement a poker game in a very abstract way: Each player's function input is an algorithmic description of his strategy, and the public function output is the corresponding simulation of a poker game. However, this is usually not what people want, and therefore we present next how to implement a "real" interactive poker game with multiple rounds based on SFE.

We already have seen how to implement randomized SFE from deterministic SFE, but still all parties learn the complete function output. However, the output of an SFE protocol can be made non-public by a technique quite similar to the randomization trick. Each party P_i just has to additionally input a secret one-time pad k_i of sufficient length, and the public SFE output can then be computed as $(y_1 \oplus k_1, \ldots, y_n \oplus k_n)$, where y_i is P_i's private output. Knowing his secret key k_i, each player P_i can decrypt his (and only his) output y_i. Thereby, we can now implement randomized SFE with non-public output, e.g. the card dealing phase in a poker game.

It finally remains to reduce a stateful multi-round game to stateless SFE. If the game state solely consists of public information, each game round can just be handled as an individual SFE instance, where the function to be evaluated depends on the current round's public game state. If the game state contains some inherently non-public information, like the players' secret cards in a poker game, we need that somehow a secret state variable s is passed on from each round to the next. Again, this can be done using one-time pads. The idea of how a game round proceeds is as follows:

- Each player P_i knows a one-time pad k_i^{old} and $\tilde{s}^{\mathsf{old}} := s^{\mathsf{old}} \oplus k_1^{\mathsf{old}} \oplus \ldots \oplus k_n^{\mathsf{old}}$, where s^{old} denotes the current game state. Further, each player P_i knows some y_i^{old}, which denotes his current view of the game (e.g. his secret cards and all public information about the game state). Note that $\tilde{s}^{\mathsf{old}}$ is an encryption of s^{old} with *all* players' secret one-time pads and thus does not reveal any information about the secret game state s^{old} to any collusion of corrupted parties.
- Each player P_i chooses his next game move x_i depending on his current game view y_i^{old} and a fresh one-time pad k_i^{new}. His next round SFE input is the tuple $(k_i^{\mathsf{old}}, k_i^{\mathsf{new}}, \tilde{s}^{\mathsf{old}}, x_i)$.
- Finally, by means of randomized SFE with non-public output as described above, the game state for the next round is computed: If all players' inputs contain the same $\tilde{s}^{\mathsf{old}}$, the function secretly decrypts the current game state s^{old}, computes the new game state s^{new} from $(s^{\mathsf{old}}, x_1, \ldots, x_n)$, and outputs $\tilde{s}^{\mathsf{new}} := s^{\mathsf{new}} \oplus k_1^{\mathsf{new}} \oplus \ldots \oplus k_n^{\mathsf{new}}$ to all players. Further, each player P_i privately receives his respectively updated game view y_i^{new}.

Note that by this procedure all players choose their moves simultaneously in each game round. If players may move only sequentially, each player's move must be handled as a separate game round.

Further note that although the above procedure perfectly hides the secret game states s^{old} and s^{new} from any collusion of malicious players, a corrupted party P_i can still flip some bits of s^{old} just by flipping the corresponding bits of k_i^{old}. Thus, players might cheat and maliciously alter the secret game state. However, this can be prevented by some error detecting encoding of the secret game state; e.g., $s^{\mathsf{old}} = (\alpha, \beta, \gamma)$, where α, β, γ are elements of some sufficiently large finite field, α contains the actual game state information, β is just uniformly random, and $\gamma = \alpha \cdot \beta$. Now, during the computation of the next game state s^{new} the game can be aborted if the secretly decrypted current game state s^{old} is not of the required form $(\alpha, \beta, \alpha \cdot \beta)$. Still, if the actual game state information α has only low entropy, this approach has one last security hole. In particular, a corrupted party P_i could mount the following attack. By flipping the corresponding bits of k_i^{old}, the β- and γ-part of the current state variable s^{old} can be altered, say β is replaced by $\beta + \beta'$ and γ is replaced by $\gamma + \gamma'$. Note that P_i can arbitrarily choose β' and γ', at least if the underlying finite field has characteristic two. But now, the protocol is *not*[1] aborted if and only if $\gamma' = \alpha \cdot \beta'$. Thus, P_i can try to guess the intermediate game state information α and verify the guess, e.g., by choosing $\beta' = 1$ and $\gamma' = \alpha$. However, this last security hole can be simply closed by appending

[1] This "not" is a correction. It was missing in the printed version.

sufficiently many bits of independent randomness to the α-part of s^{old}, so that it becomes practically unpredictable.

After all, we have seen that general SFE has not only straightforward business applications, but it even allows for secure execution of complex multi-round processes. Since SFE is such a powerful cryptographic tool, this motivates thorough investigation of its elements and related problem structures.

Security definitions. Classically, cryptographic protocols were considered secure when some specific security properties could be shown. We have already seen two such properties, namely *privacy* and *correctness*. Correctness means that the protocol implements the desired functionality, and privacy guarantees that even corrupted parties do not learn more during a protocol run than what can be inferred from their own input and output. At first glance, it might seem intuitively clear that these two properties are exactly what is usually meant by "security". On closer inspection however, things turn out pretty complicated. Giving a formal definition for the privacy property, for example, just does not work as straightforward as suggested above. Since a malicious party's behavior during a protocol run might not match any valid input value, it is initially not well-defined at all what can be inferred from this party's "input" and output. But above all, the approach of checking individual security properties yields one severe problem: How can one guarantee that a given list of security properties covers all potential attacks in every possible context? Consider as an illustrating example the secure computation of the boolean XOR operation; i.e. Alice and Bob each choose a secret input bit and the public output is the respective XOR. Obviously, Alice can always perfectly reconstruct Bob's input bit from her own input and the public output, and analogously Bob always learns Alice's input. Thus, privacy is no issue here. However, if e.g. Alice learns Bob's input *before* she chooses her own input, she has just full control over the computation result. The latter is not what one would expect from a secure XOR gate. So, in addition to privacy and correctness some kind of *independence of inputs* seems essential for secure computations.

The need for a more generic notion of security motivated several simulation based approaches [Bea92, MR92, Can01, Gol04]. The main idea there is that every possible behavior of the corrupted parties (coordinated by some adversary $\mathcal{A}$) should be imitable by a simulator $\mathcal{S}$ in an ideal model, which is secure by definition. The first simulation based notions of security only demanded for simulatability in retrospect, i.e. the adversary $\mathcal{A}$ produces output only once (at the end of the protocol) and the simulator $\mathcal{S}$ has to generate some output, such that conditioned to each possible input of the honest parties the simulator's output and the adversary's output are indistinguishable. This *stand-alone simulatability* allows for sequential protocol composition, but in case of concurrent composition there are no security guarantees any more. Security models that aim towards parallel protocol composition additionally bring an environment into play that interacts with the adversary or simulator respectively and coordinates the input behavior of the honest parties. If the environment provably cannot tell apart between real model and ideal model, the protocol is a *universally composable* implementation of the ideal functionality. Moreover, compared to stand-alone simulatability approaches, the concept of an environment machine coordinating the parties' input behavior also makes it possible to consider security of arbitrarily complex multi-round processes (like our poker example) in a more natural way.

All results of this thesis are stated and proven with respect to the *universal composability* (UC) framework of [Can01], which is usually referred to as one of the strongest commonly used security notions. See also Section 2.1 and Section 6.2.

Oblivious transfer (OT). Oblivious transfer is one of the most important primitives for SFE. Its conceptual strength, which also makes it a subject of great interest within the scope of this thesis, lies in its ability to serve as a building block for more complex cryptographic protocols. This was first demonstrated by [Yao86] in a generic construction for oblivious circuit evaluation. What is

more, OT even turned out to be complete in the sense that general SFE can be implemented from it [Kil88, GL91, CGT95, IPS08].

Oblivious transfer in its basic form was introduced by [Rab81] as a two-party primitive that models a trusted erasure channel: The sender party can enter a bit b, which is transmitted with probability $\frac{1}{2}$ to the receiver party and else replaced by a special erasure symbol $\perp$. A seemingly more useful variant of OT, $\binom{2}{1}$-OT, was introduced by [EGL85] to securely implement commitments (see below) and coin tossing. In the $\binom{2}{1}$-variant of OT, the receiver party can choose to learn exactly one out of two bits provided by the sender, while the sender does not learn which bit was chosen. This variant usually is what people refer to when speaking of "oblivious transfer". However both flavors of OT are equivalent, i.e. they can be securely implemented from each other [Cré88].

As a straightforward generalization of OT, one often also considers string-OT, where the sender's inputs are complete strings instead of single bits.

Commitments. A commitment scheme allows a sender party (e.g. Alice) to commit to a chosen bit value b and later unveil b to a receiver party (e.g. Bob), such that the receiver Bob learns nothing about b before it is unveiled, and the sender Alice cannot change b after she committed herself. Thus, a commitment protocol consists of two phases, the Commit Phase and the Unveil Phase, and between these phases the bit value is fixed but hidden. This primitive also has many applications in cryptography, especially if additional computational assumptions are used.

Contribution and structure of this thesis

This thesis contributes two results to the research area of SFE. The first contribution consists in simple but comprehensive completeness criteria for finite stateless two-party primitives. More concretely, the considered primitives are secure black boxes that can be jointly queried by two parties, have finite input and output alphabets, and do not change behavior depending on time or input history. Given any such primitive, the completeness criteria from this thesis can be used to decide efficiently whether it allows for information-theoretically secure implementation of OT (and is thus complete) or not. The other contribution of this thesis is a protocol construction that allows for information-theoretically secure implementation of OT on top of an untrusted tamper-proof hardware token.

Although both contributions have the same objective (information-theoretically secure implementation of OT), the respective techniques are completely different. Therefore, it seemed most reasonable to split up the main body of the thesis into two parts, each of which is totally self-contained and can be read independently.

Completeness criteria (Part I)

Since Kilian showed in 1988 that OT is complete in the sense that every secure multi-party computation can be realized from this primitive, cryptographers are working on reductions of OT to other primitives (cf. Section 1.1). A long-standing open question in this context is the classification of finite stateless two-party primitives (so-called "cryptogates"). Over the decades, completeness criteria have been found for deterministic cryptogates (i.e. primitives without internal randomness), noisy channels, and symmetric (i.e., both parties receive the same output) or asymmetric (i.e., only one party receives any output at all) randomized cryptogates. However, the known criteria for randomized primitives other than noisy channels only hold in presence of passive adversaries (i.e., even corrupted parties still follow the protocol). This thesis now completes this line of research by providing simple but comprehensive combinatorial completeness criteria for *all* finite stateless two-party primitives. I.e., for the first time there are completeness criteria for randomized primitives that are neither symmetric nor asymmetric (but give different outputs to the querying parties), and we overcome the limitation that previous results for randomized primitives with input from *both* parties only regarded passive adversaries. This big step is only possible by a completely novel

approach. The core element of this novel approach is the definition of an algebraic structure of "idealized cheating strategies" and its thorough investigation (q.v. sections 3.1.2–3.1.3 and 4.1–4.4). The motivation for this approach is by a rather generic protocol construction (q.v. Section 3.1.1 and Section 4.5). Utilizing estimation methods from probability theory (the Hoeffding Inequality) and real algebraic geometry (the Łojasiewicz Inequality), the security proof for this protocol scheme can be based on the gained algebraic insights (q.v. sections 3.1.4 and 4.6–4.8).

A short version of the analogous classification results for the spacial case of *deterministic* primitives appeared in [KMQ11]. The respective full version with all proofs is online available [KMQ10] but not published elsewhere yet. Part I follows the basic structure of [KMQ10], though nearly all technical details are way more complex than in the deterministic case.

Secure computation from untrusted tamper-proof hardware (Part II)

Consider the following scenario. A powerful party, henceforth called Goliath, can issue tamper-proof hardware tokens and hand them over to some other party, which we henceforth call David. Each such token has a dedicated interface, so that David can communicate with it, but the token's internal state is out of reach for David. The question now is, if in this setting OT is possible, although both parties mistrust each other. The technical difficulty to be overcome is twofold. Firstly, a corrupted Goliath can program the token maliciously. Secondly, a corrupted David can query the token on *whatever* he likes, and he can do so *whenever* he likes. Interestingly, various results in the literature (q.v. Section 5.1) demonstrate the feasibility of arbitrary secure computations, based on the tokens' tamper-proofness and the assumption that Goliath cannot communicate directly with any token in David's hands. However, all these results use additional complexity assumptions and/or a large number of tokens must be exchanged. Especially the latter is usually considered a quite severe obstacle for practical realizations, and reduction of the number of tokens has been a research objective for years. For example, [MS08] stated it as an open problem to implement a bidirectional and reusable commitment functionality, which is a strictly weaker primitive than OT, from a single token. This thesis now provides the first information-theoretically secure single-token solution, and beyond that the provided solution even allows for *asymptotically optimal* implementation of OT and commitments (q.v. Section 5.2). The basic protocol construction, on which everything else is built, is a newly developed method for verifiable affine function evaluation (q.v. Section 6.4 and Section 7.1).

A short version of these results appeared in [DKMQ11], containing a less general protocol construction, which only required a much less complicated security proof but also lacks all the optimality features presented in this thesis. A draft of Part II is online available [DKMQ12a] but not published elsewhere yet.

Some general notations

Throughout this thesis we will denote by $\mathbb{R}$ the set of real numbers and by $\mathbb{N}$ the set of all naturals including zero. If we want to exclude negative values or zero, for example, we denote that by $\mathbb{R}_{\geq 0}$ or $\mathbb{N}_{>0}$ respectively.

Random variables are denoted as bold characters, e.g. $\mathbf{x}$. We refrain from the standard approach of using capitals letters for marking random variables, since we want to use them to distinguish matrices from vectors. Thus, a random matrix is denoted as $\mathbf{M}$ and a random vector as $\mathbf{v}$, for example. We denote the probability operator by $\mathbb{P}$, i.e. a random variable $\mathbf{x}$ takes some specific value x with probability $\mathbb{P}[\mathbf{x} = x]$. The expected value of $\mathbf{x}$ is denoted by $\mathbb{E}(\mathbf{x})$. This notation of probabilities and expected values yields the least possible danger of confusion when combined with the other notations used in this thesis.

Further notations are introduced where needed. Additionally, a table of symbols is provided close to the end of this thesis.

Part I

Completeness Theorems for All Finite Stateless 2-Party Primitives

1 Introduction

Oblivious transfer was introduced in [Rab81] as a trusted erasure channel. Later, in [Cré88] it was proven to be equivalent to $\binom{2}{1}$-OT, its currently most used variant, which allows a designated receiver Bob to learn only one of two bits sent by a designated sender Alice. Since the OT primitive turned out to be complete in the sense that it allows for arbitrary secure multi-party computation [Kil88, GL91, CGT95, IPS08], for numerous primitives it has been investigated whether OT can be reduced to them. In this thesis, we exhaustively treat this question for a class of primitives that we call "finite randomized 2-party functions". Each such primitive is characterized by some finite alphabets $\Upsilon_\mathrm{A}, \Upsilon_\mathrm{B}, \Omega_\mathrm{A}, \Omega_\mathrm{B}$, a probability distribution $\mathcal{R}$ with finite support R and a mapping $f : \Upsilon_\mathrm{A} \times \Upsilon_\mathrm{B} \times R \to \Omega_\mathrm{A} \times \Omega_\mathrm{B}$. Upon input $x \in \Upsilon_\mathrm{A}$ from Alice and $y \in \Upsilon_\mathrm{B}$ from Bob, the primitive internally samples a random $r \leftarrow \mathcal{R}$, computes $(a, b) = f(x, y, r)$ and outputs a to Alice and b to Bob. Equally, one can characterize any finite randomized 2-party function by its input and output alphabets $\Upsilon_\mathrm{A}, \Upsilon_\mathrm{B}, \Omega_\mathrm{A}, \Omega_\mathrm{B}$ and a family $\{\phi_{x,y}\}_{x \in \Upsilon_\mathrm{A}, y \in \Upsilon_\mathrm{B}}$ of probability mass functions over $\Omega_\mathrm{A} \times \Omega_\mathrm{B}$, such that on input $x \in \Upsilon_\mathrm{A}$ from Alice and $y \in \Upsilon_\mathrm{B}$ from Bob the primitive with probability $\phi_{x,y}(a, b)$ outputs a to Alice and b to Bob. Regarding our work, the latter notation turns out much more convenient and therefore will be used throughout the body of this part of the thesis.

This thesis generalizes the results of [KMQ11], where the completeness question was solved for the special case of *deterministic* 2-party functions, i.e. $f(x, y, r)$ is independent of the randomness r, or alternatively $\{\phi_{x,y}\}_{x \in \Upsilon_\mathrm{A}, y \in \Upsilon_\mathrm{B}} \subseteq \{0, 1\}^{\Omega_\mathrm{A} \times \Omega_\mathrm{B}}$. Although some general ideas from the deterministic case do carry over straightforwardly, crucial techniques do not—cf. [KMQ10, Section 5]. In addition to an appropriate representation of randomized functions, we need to develop an entire tool set of technical lemmata, some of which may be of independent interest.

1.1 Related work

General related work. In the literature one finds OT protocols for bounded-classical-storage [CCM98] and bounded-quantum-storage models [DFR$^+$07] as well as noisy classical [CMW05, Wul09, IKO$^+$11] and quantum channels [Yao95, May95, May96], the latter taking commitments for granted. An entire line of research deals with implementing OT from tamper-proof hardware assumptions [BOGKW88, GKR08, CGS08, Kol10, GIMS10, GIS$^+$10, DKMQ11, CKS$^+$11]. There are reductions of $\binom{2}{1}$-OT to weaker OT versions that leak additional information [CK90, DKS99, Wul07] and to Rabin-OT [Cré88]. OT-combiners implement OT from granted sets of OTs with faulty members [MPW07, HIKN08]. For reversing the direction of $\binom{2}{1}$-OT a protocol is known with optimal number of OT queries [WW06]. Relative to computational assumptions, all-or-nothing laws have been shown [BMM99, HNRR06, MPR10], i.e. all considered non-trivial primitives are complete.

Precursory results to this work. The line of research we deal with was initiated by [Kil91], where completeness criteria for deterministic symmetric 2-party functions (i.e., both parties receive the same output, computed deterministically from their inputs) without any additional computational assumptions were provided. This line of research was continued by [Kil00], providing completeness criteria for deterministic asymmetric 2-party functions (i.e., only one party receives any meaningful output, computed deterministically from both parties' inputs). Randomized symmetric and asymmetric 2-party functions (i.e., a single output symbol, computed from both parties inputs and some secret randomness, is handed over either to both parties or only to one party) were also treated in [Kil00], but only with respect to passive adversaries (i.e., even corrupted parties still follow the

protocol). Rather recently, the completeness criteria of [Kil91, Kil00] for deterministic 2-party functions were unified and generalized by [KMQ11], now covering *all* deterministic 2-party functions, what for the first time in the literature also included 2-party functions that give different outputs to Alice and Bob. Meanwhile, [CMW05] also provided exhaustive completeness criteria with respect to active adversaries (i.e., corrupted parties may arbitrarily deviate from the protocol) for a special class of randomized asymmetric 2-party functions, namely noisy channels. This thesis now completes this line of research. The main theorem in Section 2.3 unifies and generalizes all known completeness criteria for symmetric, asymmetric, deterministic and randomized 2-party functions.

Independently of this thesis, a unified and generalized formulation of the completeness criteria from [Kil91, Kil00, CMW05, KMQ11] was found by [MPR12]. Their result is equivalent to the criteria provided by this thesis, but they only give a proof with respect to passive adversaries. Proving their conjecture for active adversaries was left as an open problem.

1.2 Our contribution

Results. We give a complete characterization of *all* finite randomized 2-party functions that allow for information-theoretically secure implementation of OT. For the reduction we provide a protocol scheme, which is universally composable—cf. [Can01]. Our characterization is based on surprisingly simple combinatorial criteria and our results are tight: Necessity of our criteria still holds, even if only correctness and privacy of the implemented OT are required. As a remarkable corollary of our work all non-complete finite 2-party functions turn out essentially symmetric.

Our work exceeds the precursory completeness criteria in two ways. Firstly, we overcome the limitation that results for randomized primitives with input from *both* parties only regarded passive adversaries. Secondly, our results also cover randomized primitives that are neither symmetric nor asymmetric (but give different meaningful outputs to Alice and Bob).

Techniques. Our starting point is a very generic protocol scheme, such that all perfectly undetectable attack strategies do comply with certain polynomial equations and hence form an algebraic variety. One major part of our work consists in finding protocol parameters, such that this algebraic variety collapses to trivial attack strategies that do not affect security at all. Using powerful tools from real algebraic geometry (namely the Łojasiewicz Inequality) and probability theory (namely the Hoeffding Inequality), we can then link real protocol runs to idealized attack strategies and thereby prove cryptographic security of our construction. This approach for protocol design and proving security might be of further interest, independently of our concrete classification results.

1.3 Organization of Part I

The basic structure of this part follows [KMQ11], though nearly all technical details in our case are way more complex. We briefly present our results in Section 2, where we first refer to the used notion of security (Section 2.1), then introduce the basic concepts needed for formulation of our results (Section 2.2), state our classification results (Section 2.3), and finally give a short overview about how our approach matches former completeness criteria in the literature (Section 2.4). In Section 3 we give an exposition of how one can prove our results. All formal proofs of our main technical contribution are located in Section 4; to make it self-contained, all needed definitions, notations and lemmata are also restated there.

2 Presentation of our results

Before we get started, we introduce two handy notations, which will make things much easier in the upcoming sections.

Finite sums of function values: Given any set T with finite subset $S \subseteq T$ and some mapping $g : T \to \mathbb{R}$, we set $g(S) := \sum_{\omega \in S} g(\omega)$ for convenience. For functions with more arguments and also for function families we use the canonical extension of this notation, e.g.:

$$\phi_{\Upsilon_A, y}(\Omega_A, b) := \sum\nolimits_{x \in \Upsilon_A,\, a \in \Omega_A} \phi_{x,y}(a, b)$$

Spaces of probability mass functions: Given some finite alphabet Ω, we denote the set of all probability mass functions over Ω by $\mathrm{pmf}(\Omega)$, i.e. $\mathrm{pmf}(\Omega) = \{\rho : \Omega \to \mathbb{R}_{\geq 0} \mid \rho(\Omega) = 1\}$.

We also use the following standard notions.

Negligibility: A function $\mu : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is negligible (in the parameter k), if $\lim_{k \to \infty} \mu(k) \cdot f(k) = 0$ for every polynomial $f \in \mathbb{R}[X]$.

Indistinguishability: Two random variables $\mathbf{x}, \mathbf{y}$ are (statistically) indistinguishable, if their statistical distance $\frac{1}{2} \sum_\alpha |\mathbb{P}[\mathbf{x} = \alpha] - \mathbb{P}[\mathbf{y} = \alpha]|$ is negligible in some security parameter.

2.1 Notion of security

Our main contribution is the construction and security proof of a generic reduction protocol that implements OT from any appropriate 2-party function. For the definition what "security" means, we lean on one of the strongest commonly used notions of security: the Universal Composability (UC) framework of [Can01]. However, our results also hold with respect to all weaker security notions that still require secure function evaluation to be private (i.e., no party can learn anything that cannot be learned from its function input and function output) and correct (i.e., if all parties follow the protocol, the desired function value is evaluated correctly).

In the UC framework, security is defined by comparison of an *ideal model* and a *real model*. The protocol of interest is running in the latter, where an adversary $\mathcal{A}$ coordinates the behavior of all corrupted parties. In the ideal model, which is secure by definition, an ideal functionality $\mathcal{F}$ implements the desired protocol task and a simulator $\mathcal{S}$ tries to mimic the actions of $\mathcal{A}$. An environment $\mathcal{Z}$ is plugged either to the ideal or the real model and has to guess, which model it is actually plugged to. When $\mathcal{Z}$ cannot distinguish between ideal and real model, the protocol is considered *UC-secure*. More formally, UC-security requires that for every adversary $\mathcal{A}$ there exists a simulator S, such that for all environments $\mathcal{Z}$ the view of $\mathcal{Z}$ in the real model (with adversary $\mathcal{A}$) is indistinguishable from the view of $\mathcal{Z}$ in the ideal model (with simulator $\mathcal{S}$). Since all our results are of information-theoretic nature, the adversarial entities $\mathcal{A}, \mathcal{S}$ and the environment $\mathcal{Z}$ are computationally unbounded (but nonetheless the running time of a simulator $\mathcal{S}$ will always be polynomial in the running time of the according adversary $\mathcal{A}$, as it is usually desired).

If the views of $\mathcal{Z}$ in the ideal model and the real model are distributed identically, we speak of *perfect* security; if there is some negligible statistical distance between these views, we have only *statistical* security. As already mentioned, one also differentiates between *passive* adversaries (i.e., corrupted parties still follow the protocol) and *active* adversaries (i.e., corrupted parties may deviate from the protocol arbitrarily). For further details see [Can01].

Since our protocol scheme implements $\binom{2}{1}$-OT from some given 2-party function, we also need a so-called *hybrid functionality* in the real model that provides access to the latter. See Figure 2.1

Functionality: $\mathcal{F}_{\text{SFE}}^{(F)}$

Let F be characterized by a family of probability mass functions $\{\phi_{x,y}\}_{x\in\Upsilon_A,y\in\Upsilon_B} \subseteq \text{pmf}(\Omega_A \times \Omega_B)$, where Υ_A, Ω_A are Alice's input and output alphabet and Υ_B, Ω_B are Bob's input and output alphabet.

- Upon receiving input (x, i) from Alice, verify that $(x, i) \in \Upsilon_A \times \mathbb{N}$ and that there is no recorded tuple $(\tilde{x}, i, \texttt{Alice})$; else ignore that input. Next, record $(x, i, \texttt{Alice})$ and send $(\texttt{processing}, \texttt{Alice}, i)$ to the adversary.
- Upon receiving input (y, i) from Bob, verify that $(y, i) \in \Upsilon_B \times \mathbb{N}$ and that there is no recorded tuple $(\tilde{y}, i, \texttt{Bob})$; else ignore that input. Next, record $(y, i, \texttt{Bob})$ and send $(\texttt{processing}, \texttt{Bob}, i)$ to the adversary.
- As soon as there are recorded tuples $(x, i, \texttt{Alice})$ and $(y, i, \texttt{Bob})$ for the same index i, generate randomly $(a, b) \in \Omega_A \times \Omega_B$ according to the distribution specified by $\phi_{x,y}$, and store (a, b, i).
- Upon receiving a message $(\texttt{Delivery}, \texttt{Alice}, i)$ from the adversary, verify that there is a stored tuple (a, b, i); else ignore that message. Next, output (a, i) to Alice and henceforth ignore all messages $(\texttt{Delivery}, \texttt{Alice}, i)$ with the same index i.
- Upon receiving a message $(\texttt{Delivery}, \texttt{Bob}, i)$ from the adversary, verify that there is a stored tuple (a, b, i); else ignore that message. Next, output (b, i) to Bob and henceforth ignore all messages $(\texttt{Delivery}, \texttt{Bob}, i)$ with the same index i.

When a party is corrupted, the adversary is granted unrestricted access to the channel between $\mathcal{F}_{\text{SFE}}^{(F)}$ and the corrupted party, including the ability of deleting and/or forging arbitrary messages; i.e., the adversary can arbitrarily send and receive messages on behalf of the corrupted party.

Figure 2.1: The ideal functionality for secure evaluation of a 2-party function F. Adapted and simplified version of the Secure Function Evaluation functionality in [Can01]. Note that via the parameter i just the same multi-session ability is achieved as in [Can01] by multiple session IDs.

for a formal definition of the hybrid functionality used. As $\binom{2}{1}$-OT itself is just a special 2-party function that on input $(b_0, b_1) \in \{0,1\}^2$ from Alice and $c \in \{0,1\}$ from Bob with probability 1 outputs b_c to Bob and a special "nothing" symbol $\perp$ to Alice, we can omit an explicit definition of the ideal OT functionality and instead use an accordingly instantiated version of the functionality from Figure 2.1.

2.2 Basic concepts

Finite randomized 2-party functions. A finite randomized 2-party function can be characterized by its input and output alphabets and output distributions (cf. Figure 2.1). By $\mathfrak{F}_{\text{fin}}$ we denote the set of all tuples $(\Upsilon_A, \Upsilon_B, \Omega_A, \Omega_B, \phi)$, where $\Upsilon_A, \Upsilon_B, \Omega_A, \Omega_B$ are non-empty finite alphabets and $\phi := \{\phi_{x,y}\}_{x\in\Upsilon_A,y\in\Upsilon_B}$ is a family of probability mass functions over $\Omega_A \times \Omega_B$, i.e. $\phi \subseteq \text{pmf}(\Omega_A \times \Omega_B)$. For convenience we will not always differentiate pedantically between the mathematical object $F \in \mathfrak{F}_{\text{fin}}$ and the corresponding primitive $\mathcal{F}_{\text{SFE}}^{(F)}$, but from the context should always be clear what is meant.

Canonical and condensed canonical representations. Our notion of $\mathfrak{F}_{\text{fin}}$ turns out a bit too detailed, since Alice and Bob can always locally relabel their input-output tuples without any side effects. For our purposes there is no need to distinguish between some $F \in \mathfrak{F}_{\text{fin}}$ and any relabeled version of F. Therefore, we introduce the concept of *canonical representations*. Given any $F \in \mathfrak{F}_{\text{fin}}$, we cannot just write down a function table for F, since each input tuple only specifies an output distribution rather than a concrete output tuple. However, for each individual input tuple we can represent the respective joint output distribution by a probability matrix with rows labeled by

		0		**1**		**2**	
		0	*1*	*0*	*1*	*0*	*1*
0	*0*	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	0	$\frac{1}{2}$
	1	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	0	$\frac{1}{2}$
1	*0*	$\frac{1}{4}$	$\frac{1}{4}$	0	$\frac{1}{3}$	0	0
	1	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{3}$	0	1

$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$
$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$
$\frac{1}{4}$	$\frac{1}{4}$		$\frac{1}{3}$	
$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{3}$	1

1	$\frac{1}{2}$	$\frac{1}{2}$	1
$\frac{1}{2}$		$\frac{1}{3}$	
$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{3}$	1

Figure 2.2: Different representations of a 2-party function that on input $x \in \{0,1\}$ from Alice and $y \in \{0,1,2\}$ from Bob outputs some uniformly random $a, b \in \{0,1\}$, subject to the conditions that $a + b \geq x \cdot y$ and $b \geq y - 1$. To the left, inputs (bold) and outputs (italic) are displayed grayed out. The matrix in the middle is a *canonical representation* of the same 2-party function with zero probabilities omitted for better readability; the right matrix is a *condensed canonical representation*.

Alice's output symbols and columns labeled by Bob's output symbols. Then, we can arrange these "inner" probability matrices in an "outer" block matrix with rows labeled by Alice's input symbols and columns labeled by Bob's input symbols (see first two tables in Figure 2.2 for an example).

Moreover, we also want to abstract from the fact that, e.g., Bob could always concatenate the result of a local coin toss to his output, thus formally doubling the size of his output alphabet just by an easily reversible local computation. Such local coin tosses appear in a canonical representation as pairwise linearly dependent columns within the same block column, or pairwise linearly dependent rows within the same block row respectively. However, we can easily get rid of them just by adding up the respective linearly dependent rows or columns. If all local coin tosses are removed from a canonical representation this way, we call it *condensed* (cf. last table in Figure 2.2).

Isomorphism. Note that the condensed canonical representation of a finite 2-party function is unique up to permutations of rows within single block rows, permutations of columns within single block columns, and permutation of rows and/or columns of the outer block matrix. Now, if two given 2-party functions $F, F' \in \mathfrak{F}_{\mathsf{fin}}$ have the same (set of) condensed canonical representations, we call them *isomorphic*. Obviously, isomorphism is an equivalence relation on $\mathfrak{F}_{\mathsf{fin}}$ and any two isomorphic 2-party functions $F, F' \in \mathfrak{F}_{\mathsf{fin}}$ can be straightforwardly implemented from each other with perfect security.

Redundancy and equivalence. Our notion of isomorphism will turn out very handy for formulation of our classification results with respect to passive adversaries, but for active adversaries we need one additional concept. In particular, there may exist input symbols that a corrupted party never needs to use, since one can always learn strictly more by inputting something else. Given any $F := (\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Omega_{\mathrm{A}}, \Omega_{\mathrm{B}}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$, we call an input symbol $y' \in \Upsilon_{\mathrm{B}}$ *redundant*, if a corrupted Bob instead of sending y' to F can always replace this input by an appropriately distributed random choice from $\Upsilon_{\mathrm{B}} \backslash \{y'\}$ and still perfectly simulate honest behavior. This is possible, if Alice's output distribution is not changed at all and Bob can reconstruct an appropriately distributed output $b' \in \Omega_{\mathrm{B}}$ from his actual input-output tuple (y, b). Formally, $y' \in \Upsilon_{\mathrm{B}}$ is *redundant*, if there exist an "input replacement strategy" $\tau \in \mathrm{pmf}(\Upsilon_{\mathrm{B}})$ and an "output reconstruction strategy" $\{\lambda_{y,b}\}_{y \in \Upsilon_{\mathrm{B}}, b \in \Omega_{\mathrm{B}}} \subseteq \mathrm{pmf}(\Omega_{\mathrm{B}})$, such that $\tau(y') = 0$ and for all $x \in \Upsilon_{\mathrm{A}}$, $a \in \Omega_{\mathrm{A}}$, $b' \in \Omega_{\mathrm{B}}$ it holds:

$$\phi_{x,y'}(a,b') = \sum\nolimits_{y \in \Upsilon_{\mathrm{B}},\, b \in \Omega_{\mathrm{B}}} \tau(y) \cdot \phi_{x,y}(a,b) \cdot \lambda_{y,b}(b')$$

For input symbols $x \in \Upsilon_{\mathrm{A}}$, redundancy is defined analogously. If neither Υ_{A} nor Υ_{B} contains any redundant input symbols, we say that F is *redundancy-free*.

W.l.o.g., malicious parties never use redundant input symbols, since they can gather exactly the same or even strictly more information by the respective input replacement and output recon-

struction strategies. Also, there is no need to constrain what honest parties may learn. Therefore, regarding active adversaries we can consider any 2-party functions *equivalent* when they only differ in some redundant input symbols. Formally, any 2-party functions $F, F' \in \mathfrak{F}_{\mathsf{fin}}$ are *equivalent*, if they can be made isomorphic by successive removal of redundant input symbols. Note that a step-by-step removal of one symbol at a time is crucial here: There may exist two input symbols that are both redundant, but after removing one of them, the other one is not redundant any more—e.g., $\Upsilon_{\mathrm{B}} = \{y, y'\}$ with $\phi_{x,y}(a,b) = \phi_{x,y'}(a,b)$ for all $x \in \Upsilon_{\mathrm{A}}$, $a \in \Omega_{\mathrm{A}}$, $b \in \Omega_{\mathrm{B}}$.

It will turn out that the *redundancy-free version* of any given $F \in \mathfrak{F}_{\mathsf{fin}}$ is unique up to isomorphism and thus equivalence of 2-party functions in the sense above is indeed an equivalence relation on $\mathfrak{F}_{\mathsf{fin}}$. However, due to lack of some required technical tools at this point, we postpone the proof to Section 4.3 (see Corollary 19).

2.3 Completeness criteria for *all* finite randomized 2-party functions

With the concepts from Section 2.2 we can now formulate our classification results. We just state the mere assertions here; for an outline of the proof see Section 3.

Definition (OT-cores)**.** Given $F := (\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Omega_{\mathrm{A}}, \Omega_{\mathrm{B}}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$, an *OT-core* of F is a non-diagonal full-rank 2×2-submatrix of the canonical representation; i.e., for the corresponding input-output tuples $(x,a),(x',a') \in \Upsilon_{\mathrm{A}} \times \Omega_{\mathrm{A}}$ and $(y,b),(y',b') \in \Upsilon_{\mathrm{B}} \times \Omega_{\mathrm{B}}$ we have the following inequation with at most one zero factor:

$$\phi_{x,y}(a,b) \cdot \phi_{x',y'}(a',b') \neq \phi_{x',y}(a',b) \cdot \phi_{x,y'}(a,b')$$

In this situation, we also call $\{(x,a),(x',a')\} \times \{(y,b),(y',b')\}$ an *OT-core* of F.

Theorem (Classification Theorem)**.** *For every $F \in \mathfrak{F}_{\mathsf{fin}}$ it holds:*

1. *OT can be implemented from $\mathcal{F}_{\mathrm{SFE}}^{(F)}$ statistically secure against* passive *adversaries, iff F has an OT-core.*
2. *OT can be implemented from $\mathcal{F}_{\mathrm{SFE}}^{(F)}$ statistically secure against* active *adversaries, iff the redundancy-free version of F has an OT-core.*

Definition (Symmetric 2-party functions)**.** A 2-party function $F := (\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Omega_{\mathrm{A}}, \Omega_{\mathrm{B}}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$ is *symmetric*, if $\phi_{x,y}(a,b) = 0$ for all $x \in \Upsilon_{\mathrm{A}}$, $y \in \Upsilon_{\mathrm{B}}$, $a \in \Omega_{\mathrm{A}}$, $b \in \Omega_{\mathrm{B}}$ with $a \neq b$.

Lemma (Symmetrization Lemma)**.** *Every 2-party function $F \in \mathfrak{F}_{\mathsf{fin}}$ that has no OT-core is isomorphic to a symmetric 2-party function.*

2.4 Comparison with criteria from the literature

The latest known completeness criteria[1] for finite 2-party functions can be subsumed by the following four theorems.

[KMQ11, Theorem 1]: A deterministic 2-party function $F := (\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Omega_{\mathrm{A}}, \Omega_{\mathrm{B}}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$ allows for implementation of OT statistically secure against *passive* adversaries, iff for the mappings $f_{\mathrm{A}} : \Upsilon_{\mathrm{A}} \times \Upsilon_{\mathrm{B}} \to \Omega_{\mathrm{A}}$ defined by $f_{\mathrm{A}}(x,y) = a :\Leftrightarrow \phi_{x,y}(a, \Omega_{\mathrm{B}}) = 1$ and $f_{\mathrm{B}} : \Upsilon_{\mathrm{A}} \times \Upsilon_{\mathrm{B}} \to \Omega_{\mathrm{B}}$ defined by $f_{\mathrm{B}}(x,y) = b :\Leftrightarrow \phi_{x,y}(\Omega_{\mathrm{A}}, b) = 1$ there exist $x, x' \in \Upsilon_{\mathrm{A}}$ and $y, y' \in \Upsilon_{\mathrm{B}}$, such that $f_{\mathrm{A}}(x,y) = f_{\mathrm{A}}(x,y')$, $f_{\mathrm{B}}(x,y) = f_{\mathrm{B}}(x',y)$, and $(f_{\mathrm{A}}(x',y), f_{\mathrm{B}}(x,y')) \neq (f_{\mathrm{A}}(x',y'), f_{\mathrm{B}}(x',y'))$.
A deterministic 2-party function $F \in \mathfrak{F}_{\mathsf{fin}}$ allows for implementation of OT statistically secure against *active* adversaries, iff its redundancy-free version allows for implementation of OT statistically secure against passive adversaries by the criterion above.

[1] Meanwhile, a unification and generalization of these criteria has been found by an independent work [MPR12]. Their criteria are equivalent to those in this thesis, but they give only a proof with respect to passive adversaries.

[CMW05, Main result]: A noisy channel allows for implementation of OT statistically secure against *active* adversaries, iff its redundancy-free version is no parallel composition of noiseless and/or capacity-zero channels.

[Kil00, Theorem 1.3]: An asymmetric $F := (\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \{\bot\}, \Omega, \phi) \in \mathfrak{F}_{\mathsf{fin}}$ allows for implementation of OT statistically secure against *passive* adversaries, iff there exist $x, x' \in \Upsilon_{\mathrm{A}}$, $y, y' \in \Upsilon_{\mathrm{B}}$ and $z, z' \in \Omega$, such that $\phi_{x,y}(\bot, z) > \phi_{x',y}(\bot, z) > 0$ or it holds that $\phi_{x,y}(\bot, z) > 0$, $\phi_{x',y}(\bot, z) > 0$, $\phi_{x,y'}(\bot, z') > 0$ and $\phi_{x',y'}(\bot, z') = 0$.

[Kil00, Theorem 1.2]: A symmetric $F := (\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Omega, \Omega, \phi) \in \mathfrak{F}_{\mathsf{fin}}$ allows for implementation of OT statistically secure against *passive* adversaries, iff there exist $x, x' \in \Upsilon_{\mathrm{A}}$, $y, y' \in \Upsilon_{\mathrm{B}}$, $z \in \Omega$, such that $\phi_{x,y}(z, z) > 0$, $\phi_{x,y'}(z, z) > 0$ and $\phi_{x,y}(z, z) \cdot \phi_{x',y'}(z, z) \neq \phi_{x,y'}(z, z) \cdot \phi_{\prime x,y}(z, z)$.

All these completeness criteria are direct corollaries of our Classification Theorem. However, the literature cited above differs substantially in the used protocol constructions and also the proof techniques. This thesis generalizes the results and techniques of [KMQ11], where also a Symmetrization Lemma were provided for deterministic 2-party functions. In particular, we adapt from [KMQ11] and generalize the notions of "redundancy" (q.v. Section 2.2), "OT-cores" (q.v. Section 2.3) and "cheating situations" (q.v. Section 3.1.2), and we also adopt the basic protocol scheme for generation of correlated data (q.v. Section 3.1.1). However, due to increased complexity the similarities are limited to a fairly abstract level. Core proof techniques of [KMQ11] are strictly bound to the deterministic case—cf. [KMQ10, Section 5]—and therefore new solutions (including a powerful lemma from real algebraic geometry, q.v. Section 4.6) are needed for randomized primitives.

3 How to prove the Classification Theorem

Necessity of our criteria. By our Symmetrization Lemma and [Kil00, Theorem 1.2] it directly follows that OT-cores are necessary for completeness with respect to passive adversaries. Moreover, the proof in [Kil00, Section 4.1] for necessity of OT-cores holds in the same way with respect to active adversaries. So, at this point we only need to give a proof for the Symmetrization Lemma.

Proof-sketch. Let some arbitrary $F := (\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Omega_{\mathrm{A}}, \Omega_{\mathrm{B}}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$ be given that has no OT-core. We have to show that F is symmetric up to isomorphism. Our first observation is that we can replace Bob's output symbols by normalized versions of the respective column vectors in the condensed canonical representation of F, i.e. upon Bob's input y we replace his function output b by the following $\mathbb{R}^{\Upsilon_{\mathrm{A}} \times \Omega_{\mathrm{A}}}$-vector:

$$\frac{1}{\phi_{\Upsilon_{\mathrm{A}},y}(\Omega_{\mathrm{A}}, b)} \cdot \left(\phi_{x,y}(a,b)\right)_{x \in \Upsilon_{\mathrm{A}}, a \in \Omega_{\mathrm{A}}}$$

Since by construction there are never any two linearly dependent columns within the same block column of a condensed canonical representation, this replacement of output symbols is an isomorphism of 2-party functions. Analogously, we can replace Alice's output symbols; let $\hat{\Omega}_{\mathrm{A}} \subseteq \mathbb{R}^{\Upsilon_{\mathrm{B}} \times \Omega_{\mathrm{B}}}$ and $\hat{\Omega}_{\mathrm{B}} \subseteq \mathbb{R}^{\Upsilon_{\mathrm{A}} \times \Omega_{\mathrm{A}}}$ denote the new output alphabets.

Now we exploit that F has no OT-core. Given any $\hat{a} \in \hat{\Omega}_{\mathrm{A}}$ and $\hat{b}, \hat{b}' \in \hat{\Omega}_{\mathrm{B}}$ with $\phi_{\Upsilon_{\mathrm{A}},\Upsilon_{\mathrm{B}}}(\hat{a}, \hat{b}) > 0$ and $\phi_{\Upsilon_{\mathrm{A}},\Upsilon_{\mathrm{B}}}(\hat{a}, \hat{b}') > 0$, it must hold that $\hat{b} = \hat{b}'$, as otherwise the two-column matrix $(\hat{b}, \hat{b}')$ would contain a non-diagonal full-rank 2×2-matrix and thereby we had an OT-core. Analogously, for all $\hat{a}, \hat{a}' \in \hat{\Omega}_{\mathrm{A}}$ and $\hat{b} \in \hat{\Omega}_{\mathrm{B}}$ with $\phi_{\Upsilon_{\mathrm{A}},\Upsilon_{\mathrm{B}}}(\hat{a}, \hat{b}) > 0$ and $\phi_{\Upsilon_{\mathrm{A}},\Upsilon_{\mathrm{B}}}(\hat{a}', \hat{b}) > 0$ it must hold that $\hat{a} = \hat{a}'$. Thus, Alice and Bob have always full information about each other's output and the function can as well announce the complete output tuple $(\hat{a}, \hat{b})$ to both of them in the first place. □

Sufficiency in the passive case. Given some $F := (\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Omega_{\mathrm{A}}, \Omega_{\mathrm{B}}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$ that has an OT-core, and given that there is only a passive adversary, we can easily implement a non-trivial noisy channel (shown to be complete in [CMW05, Wul09, IKO+11]) by the following protocol:

0. Alice and Bob agree on a bijection $\sigma : \Upsilon_{\mathrm{A}} \times \Omega_{\mathrm{A}} \to \{0, \ldots, |\Upsilon_{\mathrm{A}} \times \Omega_{\mathrm{A}}| - 1\}$. The image S of σ also serves as Alice's channel input alphabet.
1. Alice and Bob query F once with uniformly random input, thus generating input-output tuples $(x, a) \in \Upsilon_{\mathrm{A}} \times \Omega_{\mathrm{A}}$ and $(y, b) \in \Upsilon_{\mathrm{B}} \times \Omega_{\mathrm{B}}$ respectively.
2. Alice announces to Bob her intended channel input encrypted with (x, a) as follows: If she wants to send some $m \in S$ via the noisy channel, she announces $\tilde{m} := m + \sigma(x, a) \bmod |S|$.
3. Bob's noisy channel output is $(\tilde{m}, y, b)$.

Since F has an OT-core and even corrupted parties still follow the protocol, the implemented channel is not completely decomposable into noiseless channels and/or channels with zero capacity. This is straightforward to verify and suffices to implement OT by the above-mentioned literature.

Sufficiency in the active case. As we are already done with necessity of our criteria in the active and passive case and sufficiency in the passive case, so to speak "75%" of our Classification Theorem are proven. However, the vastly major part still lies ahead of us. For proving sufficiency in the active case, i.e. proving that in presence of an active adversary OT can still be reduced to any redundancy-free 2-party function that has some OT-core, we need an entire new tool set of technical lemmata and several sophisticated results from the literature. The high level idea of the reduction approach is as follows. First, Alice and Bob generate some amount of correlated data by

a)

$\frac{1}{3}$ $\frac{2}{3}$	$\frac{2}{3}$ $\frac{1}{3}$	$\frac{1}{2}$ $\frac{1}{2}$
1	1	$\frac{1}{2}$ $\frac{1}{2}$

b)

$\frac{1}{2}$ $\frac{1}{2}$	$\frac{1}{2}$ $\frac{1}{2}$	1	1
$\frac{1}{2}$ $\frac{1}{2}$	$\frac{4}{9}$ $\frac{5}{9}$	1	1

Figure 3.1: a) Example for illustration that not every OT-core is useful for us: The first two block columns contain an OT-core, but can be subsumed by the last block column.
b) Example for illustration that redundancy here is more complex than in the deterministic case: The first block column is redundant (it can be subsumed by the last two), but the second is not.

repeatedly querying the given 2-party function with random input. Within a subsequent test step each party has to partially unveil its data, so that significant cheating can be detected. Then, in a similar approach as in the passive case, the remaining data is used for implementation of non-trivial noisy channels: Alice just announces her channel inputs one-time-pad encrypted with her part of the correlated data, and Bob, since his view gives him only partial information about the used one-time pads, can only recover noisy versions of Alice's channel inputs. However, things will turn out a bit more complicated than in the passive case, since corrupted parties can try to gather some additional information by occasionally deviating from the protocol.

The first part (secure generation of correlated data, q.v. Section 3.1) is much more challenging than the second part (building OT from correlated data, q.v. Section 3.2). The former needs numerous novel techniques (see Section 4 for the formal proofs), whereas the latter mainly consists in rather straightforward adaptions of nowadays folklore techniques from the literature.

3.1 Secure generation of correlated data

In this section we explain how one can securely generate non-trivially correlated data from any redundancy-free 2-party function that has some OT-core. The main idea is to use inputs belonging to a specific OT-core with relatively high probability and all other inputs only with relatively low probability—the latter will just serve for test purposes. Notably, an all-over uniform input distribution is not suitable in general, but still all input symbols have to be used with some significant probability. We illustrate this by two examples. Our first example, given by in Figure 3.1.a, illustrates the problem with all-over uniform input distributions. In this example, a corrupted Bob can substitute a query on the first input symbol and a query on the second input symbol by two queries on the third input symbol. So, instead of uniformly choosing from his complete input alphabet, he can always input the last input symbol and thereby always get full information about Alice's input-output tuple. Our second example, given by Figure 3.1.b, illustrates that in general one cannot completely neglect all input symbols that do not belong to the chosen OT-core. In this example, if Alice only uses one of her input symbols all the time, this means that effectively we can remove one of the block rows and all of a sudden the redundancy-free version of the remaining part even has no OT-core any more.

3.1.1 The protocol for generating correlated data

Basic scheme. Basically, our protocol for generation of correlated data follows the very generic construction of [KMQ11]. It roughly proceeds as follows (for a formal description see Section 4.5).

1. **Invocation of F:** Alice and Bob query the underlying 2-party function F with random input for k times (k being the security parameter) and record their respective input-output tuples. A protocol parameter assigns what concrete input distributions are to be used.
2. **Check A:** Alice challenges Bob on some polynomial subset of the recorded data, where he has to reveal his input-output tuples. Alice aborts the protocol, if the joint distribution of

her own input-output tuples and Bob's claimed input-output tuples appears faulty. The test set is then removed from the recorded data.

3. **Check B:** This step equals the previous one with the roles of Alice and Bob interchanged.
4. **Output:** Both parties announce where they have used input symbols that were only for test purposes. All corresponding elements are removed from the recorded data. When too much of the recorded data has been deleted, the protocol is aborted; else each party outputs its remaining string of recorded input-output tuples.

At this point, a remark on the increased difficulty compared to the deterministic case seems indicated. In particular, there is one crucial difference between our scheme here and the protocol scheme of [KMQ11]. This difference is in the check steps Check A and Check B. In the scheme of [KMQ11], Alice checks in Check A that each of Bob's claimed input-output tuples (y', b') is consistent with her own respective input-output tuple (x, a) in the sense that $\phi_{x,y'}(a, b') \neq 0$, and that each of Bob's claimed input symbols occurs with the right frequency independently of her own input. This does not suffice in the randomized setting any more, as one can also see from the example in Figure 3.1.b. In this example, the redundant first block column and the non-redundant second block column differ only very slightly in their output distributions. Thus, if Alice only checked that Bob's claimed input-output tuples do not directly contradict her own input-output tuples, then Bob could substitute his second input symbol in this example right the same way he can already substitute the first input symbol. For this reason, in the check steps Check A and Check B of our protocol scheme described above each party must examine the *joint distribution* of its own input-output tuples and the other party's claimed input-output tuples.

Parameter choice. We have the following wish list to our protocol scheme:

- The challenge sets in the protocol steps Check A and Check B must be sufficiently large, so that any significant deviation from the prescribed input distributions can be detected.
- We want that even a malicious choice of the challenge sets does not substantially influence the joint distribution of the recorded input-output tuples.
- All input symbols must be used with sufficiently high probability, so that the problem illustrated in Figure 3.1.b does not emerge.
- In the last protocol step, where all data is deleted that does not belong to the chosen OT-core inputs, no corrupted party should be able to modify the recorded data's joint distribution more than by a vanishingly small amount.

Obviously, the first two objectives conflict with each other, and so do the last two. However, what might first sound like a paradox, can be achieved by a polynomially vanishing lower bound for the input probabilities and also a polynomially vanishing relative size of the challenge sets. More concretely, for every input symbol that is only for test purposes we choose an input probability of magnitude $O(k^{-\alpha})$ with constant $\alpha > 0$, and the challenge sets have size $O(k^{\frac{1}{2}+\beta})$ with constant $\beta < \frac{1}{2}$ (cf. Section 4.5)—for technical reasons we even choose $\beta < \frac{1}{6}$. Thus, there exists some constant $\varepsilon > 0$, such that $k - k^{1-\varepsilon}$ recorded input-output tuples from the first protocol step remain untouched throughout the rest of the protocol and are finally part of the output.

3.1.2 Idealized attack strategies

In the step Check A of the protocol scheme introduced in Section 3.1.1, instantiated with any $F := (\Upsilon_A, \Upsilon_B, \Omega_A, \Omega_B, \phi) \in \mathfrak{F}_{\mathsf{fin}}$, a corrupted Bob can of course try to pretend to have used another input distribution than he actually did. Analogously, a corrupted Alice can try to cheat in Check B, but for symmetry reasons it will suffice to consider the case of a corrupted Bob. We start our security considerations by introducing a very idealized notion of attack startegies. This notion comprises only perfectly undetectable attacks, but it will turn out later that *every* possible attack strategy is close to such a perfect strategy.

Cheating strategies. A *cheating strategy* of Bob is a triple (τ, λ, ω), consisting of

- an "actual input distribution" $\tau \in \mathrm{pmf}(\Upsilon_\mathrm{B})$,
- a "lying strategy" $\lambda := (\lambda_{y,b})_{y\in\Upsilon_\mathrm{B}, b\in\Omega_\mathrm{B}} \subseteq \mathrm{pmf}(\Upsilon_\mathrm{B}\times\Omega_\mathrm{B})$ in the sense that in Check A an input-output tuple (y,b) is claimed as (y',b') with probability $\lambda_{y,b}(y',b')$,
- and a "claimed input distribution" $\omega \in \mathrm{pmf}(\Upsilon_\mathrm{B})$,

such that for all $x \in \Upsilon_\mathrm{A}$, $a \in \Omega_\mathrm{A}$, $y' \in \Upsilon_\mathrm{B}$, $b' \in \Omega_\mathrm{B}$ and with $\upsilon \in \mathrm{pmf}(\Upsilon_\mathrm{A})$ denoting Alice's input distribution it holds:

$$\underbrace{\upsilon(x)\cdot\omega(y')\cdot\phi_{x,y'}(a,b')}_{\text{expected joint probability of } (x,a) \text{ and } (y',b')} \quad = \quad \underbrace{\sum\nolimits_{y\in\Upsilon_\mathrm{B},\, b\in\Omega_\mathrm{B}} \upsilon(x)\cdot\tau(y)\cdot\phi_{x,y}(a,b)\cdot\lambda_{y,b}(y',b')}_{\text{claimed joint probability of } (x,a) \text{ and } (y',b')}$$

Note that we can cancel $\upsilon(x)$ on both sides, since Alice uses her complete input alphabet and thus $\upsilon(x) > 0$ for all $x \in \Upsilon_\mathrm{A}$. I.e., Bob's cheating strategies are actually independent of Alice's input distribution; they either work for all of them or for none. Further note that ω is no arbitrarily selectable parameter but already completely fixed by τ and λ. In particular, for all $x \in \Upsilon_\mathrm{A}$, $y' \in \Upsilon_\mathrm{B}$ it holds:

$$\omega(y') \;=\; \omega(y')\cdot\phi_{x,y'}(\Omega_\mathrm{A},\Omega_\mathrm{B}) \;=\; \sum\nolimits_{y\in\Upsilon_\mathrm{B},\, b\in\Omega_\mathrm{B}} \tau(y)\cdot\phi_{x,y}(\Omega_\mathrm{A},b)\cdot\lambda_{y,b}(y',\Omega_\mathrm{B})$$

Last but not least, an easily verifiable but very important feature of cheating strategies lies in their relation to redundancy: An input symbol $y' \in \Upsilon_\mathrm{B}$ is redundant, iff there exists a cheating strategy (τ,λ,ω), such that $\tau(y') = 0$ and $\omega(y') = 1$. This directly follows from our definitions.

Cheating situations. Our notion of cheating strategies turns out a bit cumbersome for the following reason. Obviously, a corrupted Bob can follow a mixed strategy, e.g. by following half the time some cheating strategy (τ,λ,ω) and half the time some other cheating strategy (τ',λ',ω'). For the resulting cheating strategy $(\bar\tau,\bar\lambda,\bar\omega)$ it is intuitively clear that $\bar\tau = \frac{1}{2}\cdot\tau + \frac{1}{2}\cdot\tau'$ and $\bar\omega = \frac{1}{2}\cdot\omega + \frac{1}{2}\cdot\omega'$. On first glance one might also expect that $\bar\lambda = \frac{1}{2}\cdot\lambda + \frac{1}{2}\cdot\lambda'$, but this will not be true in general! E.g., if $\tau(y) = 0 < \tau'(y)$ for some $y \in \Upsilon_\mathrm{B}$, then we have that $\bar\lambda_{y,b} = \lambda'_{y,b}$ for all $b \in \Omega_\mathrm{B}$. To circumvent this inconvenience, we introduce the equivalent but more practical notion of *cheating situations*. Given Bob's cheating strategy (τ,λ,ω) and Alice's input distribution $\upsilon \in \mathrm{pmf}(\Upsilon_\mathrm{A})$, we define the corresponding cheating situation $\eta \in \mathrm{pmf}\big((\Upsilon_\mathrm{A}\times\Omega_\mathrm{A})\times(\Upsilon_\mathrm{B}\times\Omega_\mathrm{B})^2\big)$ as follows:

$$\eta\big((x,a),(y,b),(y',b')\big) := \upsilon(x)\cdot\tau(y)\cdot\phi_{x,y}(a,b)\cdot\lambda_{y,b}(y',b')$$

The intuition behind this is that instead of focusing on the cheating party's plan, we just count how often which kind of situation occurs during Check A. More precisely, $\eta((x,a),(y,b),(y',b'))$ stands for the relative frequency of the event that Alice's input-output tuple is (x,a), Bob's actual input-output tuple is (y,b), and Bob's claimed input-output tuple is (y',b'). Consequently, we can write:

$$\begin{aligned}
\eta|_\mathrm{A}(x) &:= \eta\big((x,\Omega_\mathrm{A}),(\Upsilon_\mathrm{B},\Omega_\mathrm{B}),(\Upsilon_\mathrm{B},\Omega_\mathrm{B})\big) = \upsilon(x)\\
\eta|_\mathrm{B}^{\mathsf{true}}(y) &:= \eta\big((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(y,\Omega_\mathrm{B}),(\Upsilon_\mathrm{B},\Omega_\mathrm{B})\big) = \tau(y)\\
\eta|_\mathrm{B}^{\mathsf{fake}}(y') &:= \eta\big((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(\Upsilon_\mathrm{B},\Omega_\mathrm{B}),(y',\Omega_\mathrm{B})\big) = \omega(y')
\end{aligned}$$

Our definition directly implies that every cheating situation η fulfills the following four conditions.

1. For all $x \in \Upsilon_\mathrm{A}$ it holds that $\eta|_\mathrm{A}(x) > 0$.
2. For all $x \in \Upsilon_\mathrm{A}$, $a \in \Omega_\mathrm{A}$, $y \in \Upsilon_\mathrm{B}$, $b \in \Omega_\mathrm{B}$ it holds:

$$\eta\big((x,a),(y,b),(\Upsilon_\mathrm{B},\Omega_\mathrm{B})\big) = \eta|_\mathrm{A}(x)\cdot\eta|_\mathrm{B}^{\mathsf{true}}(y)\cdot\phi_{x,y}(a,b)$$

3. For all $x \in \Upsilon_\mathrm{A}$, $a \in \Omega_\mathrm{A}$, $y' \in \Upsilon_\mathrm{B}$, $b' \in \Omega_\mathrm{B}$ it holds:

$$\eta\big((x,a),(\Upsilon_\mathrm{B},\Omega_\mathrm{B}),(y',b')\big) = \eta|_\mathrm{A}(x) \cdot \eta|_\mathrm{B}^{\mathsf{fake}}(y') \cdot \phi_{x,y'}(a,b')$$

4. For all $x \in \Upsilon_\mathrm{A}$, $a \in \Omega_\mathrm{A}$, $y, y' \in \Upsilon_\mathrm{B}$, $b, b' \in \Omega_\mathrm{B}$ with $\eta\big((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(y,b),(\Upsilon_\mathrm{B},\Omega_\mathrm{B})\big) > 0$ it holds:

$$\eta\big((x,a),(y,b),(y',b')\big) = \eta\big((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(y,b),(y',b')\big) \cdot \frac{\eta\big((x,a),(y,b),(\Upsilon_\mathrm{B},\Omega_\mathrm{B})\big)}{\eta\big((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(y,b),(\Upsilon_\mathrm{B},\Omega_\mathrm{B})\big)}$$

Note that these conditions basically are a polynomial equation system, of which we will take great advantage later. Let $\mathfrak{N}_\mathrm{B}^{(F)}$ denote the set of all $\eta \in \mathrm{pmf}\big((\Upsilon_\mathrm{A}\times\Omega_\mathrm{A})\times(\Upsilon_\mathrm{B}\times\Omega_\mathrm{B})^2\big)$ that fulfill them. We show now that actually $\mathfrak{N}_\mathrm{B}^{(F)}$ is just the set of all cheating situations. Given any $\eta \in \mathfrak{N}_\mathrm{B}^{(F)}$, we find some $\tau, \omega \in \mathrm{pmf}(\Upsilon_\mathrm{B})$ and $\lambda := (\lambda_{y,b})_{y\in\Upsilon_\mathrm{B}, b\in\Omega_\mathrm{B}} \subseteq \mathrm{pmf}(\Upsilon_\mathrm{B}\times\Omega_\mathrm{B})$, such that for all $y, y' \in \Upsilon_\mathrm{B}$, $b, b' \in \Omega_\mathrm{B}$ we have:

$$\begin{aligned}
\tau(y) &= \eta|_\mathrm{B}^{\mathsf{true}}(y) \\
\omega(y') &= \eta|_\mathrm{B}^{\mathsf{fake}}(y') \\
\lambda_{y,b}(y',b') &= \frac{\eta\big((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(y,b),(y',b')\big)}{\eta\big((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(y,b),(\Upsilon_\mathrm{B},\Omega_\mathrm{B})\big)} \quad \text{if } \eta\big((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(y,b),(\Upsilon_\mathrm{B},\Omega_\mathrm{B})\big) > 0
\end{aligned}$$

Since η meets the four conditions above, we have that (τ, λ, ω) is a cheating strategy:

$$\begin{aligned}
&\textstyle\sum_{y\in\Upsilon_\mathrm{B},\, b\in\Omega_\mathrm{B}} \tau(y)\cdot\phi_{x,y}(a,b)\cdot\lambda_{y,b}(y',b') \;=\; \sum_{y\in\Upsilon_\mathrm{B},\, b\in\Omega_\mathrm{B}} \frac{\eta\big((x,a),(y,b),(\Upsilon_\mathrm{B},\Omega_\mathrm{B})\big)\cdot\lambda_{y,b}(y',b')}{\eta|_\mathrm{A}(x)} \\
&= \tfrac{1}{\eta|_\mathrm{A}(x)} \cdot \textstyle\sum_{y\in\Upsilon_\mathrm{B},\, b\in\Omega_\mathrm{B}} \eta\big((x,a),(y,b),(y',b')\big) \;=\; \omega(y')\cdot\phi_{x,y'}(a,b')
\end{aligned}$$

Likewise, η is a corresponding cheating situation:

$$\eta|_\mathrm{A}(x)\cdot\tau(y)\cdot\phi_{x,y}(a,b)\cdot\lambda_{y,b}(y',b') = \eta\big((x,a),(y,b),(\Upsilon_\mathrm{B},\Omega_\mathrm{B})\big)\cdot\lambda_{y,b}(y',b') = \eta\big((x,a),(y,b),(y',b')\big)$$

Advantages of our notion of cheating situations. In contrast to the more intuitive notion of cheating strategies, our definition of cheating situations enjoys some very handy structure: When we fix Alice's input distribution, the remaining set of cheating situations is a bounded convex polytope in the affine space $\mathbb{R}^{(\Upsilon_\mathrm{A}\times\Omega_\mathrm{A})\times(\Upsilon_\mathrm{B}\times\Omega_\mathrm{B})^2}$, spanned by finitely many vertices (cf. Lemma 10). Furthermore, cheating situations inherit two important features from cheating strategies. Firstly, cheating situations can be considered independent of (honest) Alice's input distribution, since they can be rescaled canonically to any input distribution that assigns non-zero probability to every $x \in \Upsilon_\mathrm{A}$ (q.v. Lemma 8). Secondly, an input symbol $y' \in \Upsilon_\mathrm{B}$ is redundant, iff there exists a cheating situation $\eta \in \mathfrak{N}_\mathrm{B}^{(F)}$, such that $\eta|_\mathrm{B}^{\mathsf{true}}(y') = 0$ and $\eta|_\mathrm{B}^{\mathsf{fake}}(y') = 1$. For redundancy of y' it even suffices that $\eta|_\mathrm{B}^{\mathsf{true}}(y') < \eta|_\mathrm{B}^{\mathsf{fake}}(y')$ and $\eta|_\mathrm{B}^{\mathsf{true}}(y) \geq \eta|_\mathrm{B}^{\mathsf{fake}}(y)$ for all $y \in \Upsilon_\mathrm{B}\backslash\{y'\}$. This results from some useful decomposability features of the algebraic structure $\mathfrak{N}_\mathrm{B}^{(F)}$, but for now we skip all the technical details and instead just refer to Section 4.3.

Last but not least, cheating situations are also unaffected by another disadvantage of cheating strategies that misleads intuition: If $\lambda_{y,b}(y',b') > 0$, this does not necessarily mean that Bob ever really replaces an input-output tuple (y,b) by (y',b'); as well, it might be the case that $\tau(y) = 0$ (i.e., Bob did not use the input symbol y at all). In contrast, if $\eta\big((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(y,b),(y',b')\big) > 0$, then Bob has in fact replaced the corresponding portion of actual input-output tuples (y,b) by claimed input-output tuples (y',b').

3.1.3 Robust OT-cores

We aim at an instantiation of the protocol scheme described in Section 3.1.1, such that the inputs belonging to some chosen OT-core of the underlying 2-party function F are used with relatively high probability and all other inputs have relatively low probability. However, if $\tilde{y}, \tilde{y}'$ are Bob's OT-core inputs and there exists a cheating situation $\eta \in \mathfrak{N}_{\mathrm{B}}^{(F)}$, such that $\eta|_{\mathrm{B}}^{\mathsf{true}}(\{\tilde{y}, \tilde{y}'\}) = 0$ and $\eta|_{\mathrm{B}}^{\mathsf{fake}}(\{\tilde{y}, \tilde{y}'\}) = 1$, then we have no security guarantee (cf. Figure 3.1.a). We need at least that $\eta|_{\mathrm{B}}^{\mathsf{true}}(\{\tilde{y}, \tilde{y}'\}) = 1$ for all $\eta \in \mathfrak{N}_{\mathrm{B}}^{(F)}$ with $\eta|_{\mathrm{B}}^{\mathsf{fake}}(\{\tilde{y}, \tilde{y}'\}) = 1$; otherwise a corrupted Bob can always substitute a substantial fraction of his OT-core queries by other inputs. Surprisingly, this is not only a necessary precondition for security, but it will even turn out sufficient. The key idea is to choose protocol parameters, such that the prescribed probability of non-OT-core inputs is high enough for cheating detection, but still so small that only cheating strategies (τ, λ, ω) with $\omega(y) = 0$ for all non-OT-core inputs y may work. This might first sound like a paradox, but can be achieved by polynomially vanishing probabilities for the non-OT-core inputs (cf. Section 3.1.1).

However, first and foremost we need to show that there always exists an OT-core fulfilling the above-mentioned criterion, if only the redundancy-free version of the considered 2-party function has any OT-core at all (see Figure 3.1.a for a negative example). Moreover, we also need analogous security against a possibly cheating Alice, and we must rule out that every OT-core found secure against a cheating Bob is insecure against a cheating Alice and vice versa. We achieve this all at once by the next lemma (cf. Lemma 25).

Lemma. *Let some $F := (\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Omega_{\mathrm{A}}, \Omega_{\mathrm{B}}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$ be given that is redundancy-free and has an OT-core. Then there also exists an OT-core within the same rows of the canonical representation of F, such that for Bob's corresponding input symbols $\bar{y}, \bar{y}'$ and every cheating situation $\eta \in \mathfrak{N}_{\mathrm{B}}^{(F)}$ with $\eta|_{\mathrm{B}}^{\mathsf{fake}}(\{\bar{y}, \bar{y}'\}) = 1$ we have that $\eta|_{\mathrm{B}}^{\mathsf{true}}(\{\bar{y}, \bar{y}'\}) = 1$.*

By this lemma, given any OT-core, we find an OT-core within the same rows of the canonical representation, such that this new OT-core is secure against a potentially cheating Bob. Analogously, starting from the OT-core secure against Bob, we find an OT-core within the same columns of the canonical representation, such that this new OT-core is secure against a potentially cheating Alice. Since in the second step Bob's involved input symbols stay the same, the finally found OT-core is also still secure against a cheating Bob.

Now, we give a proof-sketch for this lemma, which is a core element of our line of argument. Note that, although our notion of cheating situations can be seen as a generalization of the corresponding concept in [KMQ11], this proof is independent—cf. [KMQ10, Section 5].

Proof-sketch. Let $(\tilde{x}, \tilde{a}), (\tilde{x}', \tilde{a}') \in \Upsilon_{\mathrm{A}} \times \Omega_{\mathrm{A}}$ and $(\tilde{y}, \tilde{b}), (\tilde{y}', \tilde{b}') \in \Upsilon_{\mathrm{B}} \times \Omega_{\mathrm{B}}$ denote Alice's and Bob's input-output tuples belonging to the initially given OT-core. W.l.o.g., $\phi_{\tilde{x},\tilde{y}}(\tilde{a}, \tilde{b}) > 0$ and $\phi_{\tilde{x}',\tilde{y}}(\tilde{a}', \tilde{b}) > 0$.

If $\tilde{y} = \tilde{y}'$, i.e. the initially given OT-core lies within a single block column of the canonical representation, then existence of a cheating situation $\eta \in \mathfrak{N}_{\mathrm{B}}^{(F)}$ with $\eta|_{\mathrm{B}}^{\mathsf{fake}}(\{\bar{y}, \bar{y}'\}) = 1 > \eta|_{\mathrm{B}}^{\mathsf{true}}(\{\bar{y}, \bar{y}'\})$ would imply that the input symbol simultaneously denoted by $\tilde{y}$ and $\tilde{y}'$ is redundant (cf. Corollary 18). So, in this case things are easy. Else, i.e. if $\tilde{y} \neq \tilde{y}'$, we need a more abstract view of cheating situations to keep arguments traceable. Let $\mathfrak{X}_{\mathrm{B}}^{(F)}$ denote the set of all mappings $\xi : \Upsilon_{\mathrm{B}} \to \mathbb{R}$ for that there exist some $\eta \in \mathfrak{N}_{\mathrm{B}}^{(F)}$ and $\gamma \in \mathbb{R}_{>0}$, such that $\gamma \cdot \xi(y) = \eta|_{\mathrm{B}}^{\mathsf{fake}}(y) - \eta|_{\mathrm{B}}^{\mathsf{true}}(y)$ for all $y \in \Upsilon_{\mathrm{B}}$. The intuition behind this is merely that $\xi(y) > 0$ if Bob claims to have input y more often than he actually did, and $\xi(y) < 0$ if Bob claims to have input y less often than he actually did. We will make use of the following properties of this notation:

- The set $\mathfrak{X}_{\mathrm{B}}^{(F)}$ is closed under positive linear combination, i.e. $\gamma \cdot \xi + \gamma' \cdot \xi' \in \mathfrak{X}_{\mathrm{B}}^{(F)}$ for all $\gamma, \gamma' \in \mathbb{R}_{>0}$ and $\xi, \xi' \in \mathfrak{X}_{\mathrm{B}}^{(F)}$. This directly follows from the fact that Alice's input distribution $\eta|_{\mathrm{A}}$ of every cheating situation $\eta \in \mathfrak{N}_{\mathrm{B}}^{(F)}$ can be canonically rescaled, and the fact that the set of all cheating situations with the same input distribution of Alice is convex.

	$\tilde{y}$		$\tilde{y}'$	$\tilde{Y}$							$\tilde{Y}'$							$\tilde{Y}_0$			
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
⋯	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{6}$	$\frac{1}{2}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{24}$	$\frac{1}{12}$	$\frac{1}{8}$	$\frac{1}{6}$	$\frac{1}{8}$	$\frac{5}{9}$	1	$\frac{1}{4}$	0	0	0	0	0	⋯
⋯	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{24}$	$\frac{1}{12}$	$\frac{1}{8}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{4}{9}$	0	0	$\frac{1}{3}$	$\frac{1}{4}$	0	0	0	⋯
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Figure 3.2: Illustration of the construction of the input sets $\tilde{Y}, \tilde{Y}', \tilde{Y}_0$. Input symbols from $\tilde{Y}$ have an OT-core together with $\tilde{y}'$, but not with $\tilde{y}$; inputs from $\tilde{Y}'$ have an OT-core together with $\tilde{y}$; inputs from $\tilde{Y}_0$ cannot be completed by $\tilde{y}$ or $\tilde{y}'$ to have an OT-core. Note that always $\tilde{y} \in \tilde{Y}$ and $\tilde{y}' \in \tilde{Y}'$, which is not displayed in order to keep the graphic simple.

- If for some $y' \in \Upsilon_\mathrm{B}$ there exists a $\xi \in \mathfrak{X}_\mathrm{B}^{(F)}$ with $\xi(y') > 0$ and $\xi(y) \leq 0$ for all $y \in \Upsilon_\mathrm{B} \backslash \{y'\}$, then y' is redundant. This is just a reformulation of the redundancy criterion that there exists an $\eta \in \mathfrak{N}_\mathrm{B}^{(F)}$ with $\eta|_\mathrm{B}^{\mathsf{true}}(y') < \eta|_\mathrm{B}^{\mathsf{fake}}(y')$ and $\eta|_\mathrm{B}^{\mathsf{true}}(y) \geq \eta|_\mathrm{B}^{\mathsf{fake}}(y)$ for all $y \in \Upsilon_\mathrm{B} \backslash \{y'\}$.

Further, for any $Y \subseteq \Upsilon_\mathrm{B}$ let $\Psi_F(Y)$ denote the set of all input symbols y that a corrupted Bob can use although the protocol prescribes to use only input symbols from Y, i.e.:

$$\Psi_F(Y) \;=\; \{y \in \Upsilon_\mathrm{B} \mid \text{there exists an } \eta \in \mathfrak{N}_\mathrm{B}^{(F)}, \text{ such that } \eta|_\mathrm{B}^{\mathsf{true}}(y) > 0 \text{ and } \eta|_\mathrm{B}^{\mathsf{fake}}(Y) = 1\}$$

Note that by the convex combinability of cheating situations we always find some $\eta \in \mathfrak{N}_\mathrm{B}^{(F)}$, such that $\eta|_\mathrm{B}^{\mathsf{fake}}(Y) = 1$ and $\eta|_\mathrm{B}^{\mathsf{true}}(y) > 0$ for *all* $y \in \Psi_F(Y)$. Thus, we also always have some $\xi \in \mathfrak{X}_\mathrm{B}^{(F)}$, such that $\xi(y) = 0$ for all $y \notin \Psi_F(Y)$ and $\xi(y) < 0$ for all $y \in \Psi_F(Y) \backslash Y$. Further note that always $\Psi_F(Y) \subseteq \Psi_F(Y')$ for all $Y \subseteq Y'$, that $\Psi_F(\Psi_F(Y')) = \Psi_F(Y')$, and that hence $\Psi_F(Y) \subseteq \Psi_F(Y')$ for all $Y \subseteq \Psi_F(Y')$ (cf. Lemma 23).

Now we can start with our argumentation. First of all, we divide $\Psi_F(\tilde{y}, \tilde{y}')$ into the following three subsets (cf. Figure 3.2):

- Let $\tilde{Y}'$ denote the set of all $y' \in \Psi_F(\tilde{y}, \tilde{y}')$, such that for some $b' \in \Omega_\mathrm{B}$ the $\{(\tilde{x}, \tilde{a}), (\tilde{x}', \tilde{a}')\} \times \{(\tilde{y}, \tilde{b}), (y', b')\}$-submatrix of the canonical representation of F is an OT-core; i.e., since by assumption $\phi_{\tilde{x},\tilde{y}}(\tilde{a}, \tilde{b}) > 0$ and $\phi_{\tilde{x}',\tilde{y}}(\tilde{a}', \tilde{b}) > 0$, we just need:

$$\phi_{\tilde{x},\tilde{y}}(\tilde{a}, \tilde{b}) \cdot \phi_{\tilde{x}',y'}(\tilde{a}', b') \neq \phi_{\tilde{x}',\tilde{y}}(\tilde{a}', \tilde{b}) \cdot \phi_{\tilde{x},y'}(\tilde{a}, b')$$

- Let $\tilde{Y}$ denote the set of all $y \in \Psi_F(\tilde{y}, \tilde{y}') \backslash \tilde{Y}'$, such that for some $b \in \Omega_\mathrm{B}$ the $\{(\tilde{x}, \tilde{a}), (\tilde{x}', \tilde{a}')\} \times \{(y, b), (\tilde{y}', \tilde{b}')\}$-submatrix of the canonical representation of F is an OT-core; i.e., $\phi_{\tilde{x},y}(\tilde{a}, b) > 0$ and $\phi_{\tilde{x}',y}(\tilde{a}', b) > 0$ and for all $\hat{b} \in \Omega_\mathrm{B}$ we have:

$$\phi_{\tilde{x},\tilde{y}}(\tilde{a}, \tilde{b}) \cdot \phi_{\tilde{x}',y}(\tilde{a}', \hat{b}) = \phi_{\tilde{x},y}(\tilde{a}, \hat{b}) \cdot \phi_{\tilde{x}',\tilde{y}}(\tilde{a}', \tilde{b})$$

- Let $\tilde{Y}_0$ denote the set of all $y_0 \in \Psi_F(\tilde{y}, \tilde{y}')$, such that for all $b_0 \in \Omega_\mathrm{B}$ neither the $\{(\tilde{x}, \tilde{a}), (\tilde{x}', \tilde{a}')\} \times \{(\tilde{y}, \tilde{b}), (y_0, b_0)\}$-submatrix nor the $\{(\tilde{x}, \tilde{a}), (\tilde{x}', \tilde{a}')\} \times \{(y_0, b_0), (\tilde{y}', \tilde{b}')\}$-submatrix of the canonical representation of F is an OT-core; i.e., $\phi_{\tilde{x},y_0}(\tilde{a}, \Omega_\mathrm{B}) = \phi_{\tilde{x}',y_0}(\tilde{a}', \Omega_\mathrm{B}) = 0$.

Our proof is by contradiction and hence w.l.o.g. we assume that $\Psi_F(y, \tilde{y}') = \Psi_F(\tilde{y}, \tilde{y}')$ for all $y \in \tilde{Y}$ and $\Psi_F(\tilde{y}, y') = \Psi_F(\tilde{y}, \tilde{y}')$ for all $y' \in \tilde{Y}'$—keep in mind that $\Psi_F(Z) \subseteq \Psi_F(\tilde{y}, \tilde{y}')$ for all $Z \subseteq \Psi_F(\tilde{y}, \tilde{y}')$ as mentioned above. Now we pick some arbitrary $y' \in \Psi_F(\tilde{y}, \tilde{y}')$; w.l.o.g. $y' \in \tilde{Y}'$. By assumption we find some $\xi', \xi'' \in \mathfrak{X}_\mathrm{B}^{(F)}$, such that for all $y \in \Upsilon_\mathrm{B}$ it holds:

$$\begin{array}{ll} \xi'(y) > 0 \text{ if } y \in \{\tilde{y}, y'\} & \xi''(y) > 0 \text{ if } y \in \{\tilde{y}, \tilde{y}'\} \\ \xi'(y) = 0 \text{ if } y \notin \Psi_F(\tilde{y}, \tilde{y}') & \xi''(y) = 0 \text{ if } y \notin \Psi_F(\tilde{y}, \tilde{y}') \\ \xi'(y) < 0 \text{ if } y \in \Psi_F(\tilde{y}, \tilde{y}') \backslash \{\tilde{y}, y'\} & \xi''(y) < 0 \text{ if } y \in \Psi_F(\tilde{y}, \tilde{y}') \backslash \{\tilde{y}, \tilde{y}'\} \end{array}$$

Let $\xi := \xi'(y') \cdot \xi'' - \xi''(y') \cdot \xi'$, whereby for all $y \in \Upsilon_B$ we get:

$$\begin{aligned}
&\xi(y) > 0 \text{ if } y = \tilde{y} \\
&\xi(y) = 0 \text{ if } y \notin \Psi_F(\tilde{y}, \tilde{y}') \text{ or } y = y' \\
&\xi(y) < 0 \text{ if } y \in \Psi_F(\tilde{y}, \tilde{y}') \backslash \{\tilde{y}, \tilde{y},' y'\}
\end{aligned}$$

Moreover, it must hold that $\xi(\tilde{y}') > 0$, since otherwise $\xi(y) \leq 0$ for all $y \in \Upsilon_B \backslash \{\tilde{y}\}$ and hence $\tilde{y}$ would be redundant. Iteration of this construction yields some $\hat{\xi} \in \mathfrak{X}_B^{(F)}$, such that for all $y \in \Upsilon_B$ we have:

$$\begin{aligned}
&\hat{\xi}(y) > 0 \text{ if } y \in \{\tilde{y}, \tilde{y}'\} \\
&\hat{\xi}(y) = 0 \text{ if } y \notin \{\tilde{y}, \tilde{y}'\} \cup \tilde{Y}_0 \\
&\hat{\xi}(y) < 0 \text{ if } y \in \tilde{Y}_0
\end{aligned}$$

Switching back to cheating situations or cheating strategies respectively, this means that Bob can use his input symbols $\tilde{y}, \tilde{y}'$ less frequently than prescribed and substitute them by input symbols from $\tilde{Y}_0$. However, this cannot be (the following arguments can probably be followed best through a concrete example, e.g. Figure 3.2): Since $\phi_{\tilde{x},Y_0}(\tilde{a}, \Omega_B) = 0$ and $\phi_{\tilde{x}',Y_0}(\tilde{a}', \Omega_B) = 0$, but $\phi_{\tilde{x},\tilde{y}}(\tilde{a}, \tilde{b}) > 0$ and $\phi_{\tilde{x}',\tilde{y}}(\tilde{a}', \tilde{b}) > 0$ and also $\phi_{\tilde{x},\tilde{y}'}(\tilde{a}, \tilde{b}') + \phi_{\tilde{x}',\tilde{y}'}(\tilde{a}', \tilde{b}') > 0$, this substantially decreases Alice's overall frequency of input-output tuples $(\tilde{x}, \tilde{a})$ and $(\tilde{x}', \tilde{a}')$ and thus cannot be an undetectable cheating strategy. □

3.1.4 Robust OT-cores in real protocol runs

In this section we consider real protocol runs of the protocol scheme introduced in Section 3.1.1, instantiated with some arbitrary $F := (\Upsilon_A, \Upsilon_B, \Omega_A, \Omega_B, \phi) \in \mathfrak{F}_{\mathsf{fin}}$. Unfortunately, there is no guarantee that a corrupted Bob always follows *exactly* a cheating strategy in the idealized sense of Section 3.1.2. For instance, he can as well try to exploit that Alice has to tolerate some statistical noise in the protocol step Check A. However, we show now that indeed our notion of cheating situations is a very suitable approximation of what may happen during a real protocol run. To make formulas more readable, we use the following self-suggesting notation.

Notation (Almost equality). For any $a, b, c \in \mathbb{R}$, by "$a = b \pm c$" we denote that $|a - b| \leq c$.

Linking real protocol runs to idealized attack strategies. Our starting point for linking real protocol runs to idealized attack strategies is the Hoeffding Inequality. We need it in the following form, which directly follows by [Hoe63, Theorem 1].

Lemma (Hoeffding Inequality). *Let any $n \in \mathbb{N}$, $c \in \mathbb{R}_{>0}$ and a binomially distributed random variable $\mathbf{x}$ with expected value $\mathbb{E}(\mathbf{x})$ be given. Further let $\mathbb{P}[0 \leq \mathbf{x} \leq n] = 1$. Then it holds:*

$$\mathbb{P}\left[|\mathbf{x} - \mathbb{E}(\mathbf{x})| \geq c\right] \;\leq\; 2 \cdot \exp\left(\tfrac{-2c^2}{n}\right)$$

By [Hoe63, Section 6], this lemma also holds true if $\mathbf{x}$ is distributed hypergeometrically.

Following [KMQ10, Lemma 15], we instantiate the Hoeffding Inequality with $n := k$ and $c := k^\Delta$, where k denotes our security parameter and $\Delta > \frac{1}{2}$ is constant. Thereby we get that the probability $\mathbb{P}\left[|\mathbf{x} - \mathbb{E}(\mathbf{x})| \geq k^\Delta\right]$ is upper bounded by $2 \cdot \exp(-2k^{2\Delta-1})$. I.e., it vanishes exponentially in k and hence is negligible, or in other words, $\mathbf{x} = \mathbb{E}(\mathbf{x}) \pm k^\Delta$ with overwhelming probability.

The most apparent application of the Hoeffding Inequality is Alice's choice of the challenge set in the protocol step Check A. This random choice is a hypergeometric sampling process and by the hypergeometric version of the Hoeffding Inequality it follows that the joint distribution of Alice's and Bob's input-output tuples in the challenge set is a good approximation of their overall

joint distribution of input-output tuples. Moreover, for any $x \in \Upsilon_A$, $a \in \Omega_A$, $y \in \Upsilon_B$, $b \in \Omega_B$ and with $\upsilon \in \mathrm{pmf}(\Upsilon_A)$ denoting Alice's input distribution it holds: Whenever in the first protocol step Bob inputs y, then with probability $\upsilon(x) \cdot \phi_{x,y}(a,b)$ this counts to the number of events where Alice's input-output tuple is (x,a) and Bob's input-output tuple is (y,b). Therefore, Bob's input strategy can be seen as a binomial sampling process, and thus the binomial version of the Hoeffding Inequality applies. Analogously, the hypergeometric version of the Hoeffding Inequality applies to Bob's lying strategy in the step Check A. Skipping some further details, in the end this yields: If for each $(x,a,y,b,y',b') \in \Upsilon_A \times \Omega_A \times \Upsilon_B \times \Omega_B \times \Upsilon_B \times \Omega_B$ we count the relative frequency of the event that Alice's input-output tuple is (x,a), Bob's actual input-output tuple is (y,b) and Bob's claimed input-output tuple is (y',b'), then with overwhelming probability the resulting $\mathbb{R}^{(\Upsilon_A \times \Omega_A) \times (\Upsilon_B \times \Omega_B)^2}$-vector ν fulfills the defining conditions of cheating strategies as introduced in Section 3.1.2, up to some error of magnitude $k^{-\varepsilon}$ with constant $\varepsilon > 0$ (cf. Lemma 34). In particular, with $\upsilon \in \mathrm{pmf}(\Upsilon_A)$ denoting Alice's prescribed input distribution, we have:

1. For all $x \in \Upsilon_A$ it holds that $\nu|_A(x) := \nu\big((x,\Omega_A),(\Upsilon_B,\Omega_B),(\Upsilon_B,\Omega_B)\big) = \upsilon(x) \pm k^{-\varepsilon}$.
2. For all $x \in \Upsilon_A$, $a \in \Omega_A$, $y \in \Upsilon_B$, $b \in \Omega_B$ it holds:
$$\nu\big((x,a),(y,b),(\Upsilon_B,\Omega_B)\big) = \nu|_A(x) \cdot \nu\big((\Upsilon_A,\Omega_A),(y,\Omega_B),(\Upsilon_B,\Omega_B)\big) \cdot \phi_{x,y}(a,b) \pm k^{-\varepsilon}$$
3. For all $x \in \Upsilon_A$, $a \in \Omega_A$, $y' \in \Upsilon_B$, $b' \in \Omega_B$ it holds:
$$\nu\big((x,a),(\Upsilon_B,\Omega_B),(y',b')\big) = \nu|_A(x) \cdot \nu\big((\Upsilon_A,\Omega_A),(\Upsilon_B,\Omega_B),(y',\Omega_B)\big) \cdot \phi_{x,y'}(a,b') \pm k^{-\varepsilon}$$
4. For all $x \in \Upsilon_A$, $a \in \Omega_A$, $y,y' \in \Upsilon_B$, $b,b' \in \Omega_B$ with $\nu\big((\Upsilon_A,\Omega_A),(y,b),(\Upsilon_B,\Omega_B)\big) > 0$ it holds:
$$\nu\big((x,a),(y,b),(y',b')\big) = \nu\big((\Upsilon_A,\Omega_A),(y,b),(y',b')\big) \cdot \frac{\nu\big((x,a),(y,b),(\Upsilon_B,\Omega_B)\big)}{\nu\big((\Upsilon_A,\Omega_A),(y,b),(\Upsilon_B,\Omega_B)\big)} \pm k^{-\varepsilon}$$

The last three items[1] above can be seen as a polynomial equation system over $\mathbb{R}^{(\Upsilon_A \times \Omega_A) \times (\Upsilon_B \times \Omega_B)^2}$, such that the defining multivariate polynomials solely depend on F, the cheating situations from Section 3.1.2 are always in the zero locus of these polynomials, and all these polynomials evaluate on ν to something bounded by $k^{-\varepsilon}$. Now, we are going to exploit the latter and derive an estimation for the distance of our $\mathbb{R}^{(\Upsilon_A \times \Omega_A) \times (\Upsilon_B \times \Omega_B)^2}$-vector ν from $\mathfrak{N}_B^{(F)}$. This is where real algebraic geometry comes into play.

Lemma (Łojasiewicz Inequality [Łoj59, Theorem 17]). *Let some $n \in \mathbb{N}$, an open set $U \subseteq \mathbb{R}^n$, a compact set $K \subset U$, and a real analytic function $h : U \to \mathbb{R}$ with non-empty zero locus Z be given. Then, there exist some constants $c,d \in \mathbb{R}_{>0}$, such that for all $\nu \in K$ it holds:*
$$\inf_{\eta \in Z} \|\nu - \eta\| \;\leq\; c \cdot \big|h(\nu)\big|^d$$

Unfortunatley, the Łojasiewicz Inequality is not directly applicable in our case. The primary reason is that each cheating strategy $\eta \in \mathfrak{N}_B^{(F)}$ does not only have to fulfill the above-mentioned polynomial equations (which translates to $\eta \in Z$ in terms of the Łojasiewicz Inequality), but it must also hold that $\eta \in \mathrm{pmf}\big((\Upsilon_A \times \Omega_A) \times (\Upsilon_B \times \Omega_B)^2\big)$. Therefore, we needed to develop the following non-obvious adaption of the Łojasiewicz Inequality (q.v. Lemma 35).

Lemma. *Let $n \in \mathbb{N}$ and some polynomial $f \in \mathbb{R}[X_1,\ldots,X_n]$ be given, such that the variety $V := \big\{\nu \in \mathbb{R}^n \;\big|\; f(\nu) = 0\big\}$ is not empty. Furthermore, let a bounded convex polytope $P \subset \mathbb{R}^n$ be given, such that $V \cap P \neq \emptyset$. Then for every norm there exist some constants $c,d \in \mathbb{R}_{>0}$, such that for all $\nu \in P$ it holds:*
$$\min_{\eta \in V \cap P} \|\nu - \eta\| \;\leq\; c \cdot \big|f(\nu)\big|^d$$

[1] We shall just ignore the first item in this more intuitive overview. It will be formally needed to bound $\min_{x \in \Upsilon_A} \nu|_A(x)$ away from zero, but it plays a somewhat special role, since Alice's input distribution υ depends on the security parameter k (cf. Section 3.1.1). All other equations are independent of k, except for the respective error terms $k^{-\varepsilon}$.

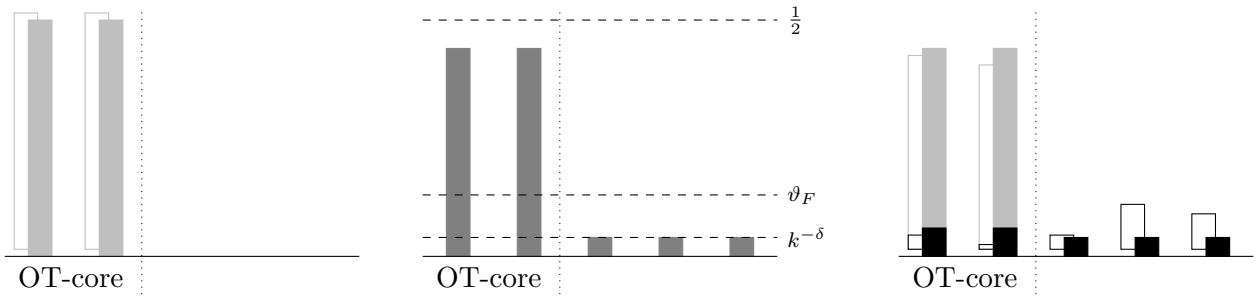

Figure 3.3: What we know (left), what we have (middle), and what we can conclude (right). Filled bars stand for claimed input probabilities, non-filled bars stand for actual input probabilities.
Left diagram: When Bob claims to have used only inputs that belong to the chosen OT-core, we know that he actually has done so.
Middle diagram: In real protocol runs we must tolerate that Bob sometimes claims to have used inputs not belonging to the chosen OT-core.
Right diagram: Decomposition of the claimed input distribution from the middle diagram into a large part, where the guarantee from the left diagram applies, and a polynomially vanishing rest.

We instantiate this lemma with $P := \mathrm{pmf}((\Upsilon_\mathrm{A}\times\Omega_\mathrm{A})\times(\Upsilon_\mathrm{B}\times\Omega_\mathrm{B})^2)$ and $f := \sum_{p\in S} p^2$, where the set S just contains the polynomials from our polynomial equation system above. Thereby, we can show that with overwhelming probability either the considered protocol run is aborted or the vector ν described above is $(c\cdot|S|\cdot k^{-2\varepsilon d})$-close to a cheating strategy $\eta \in \mathfrak{N}_\mathrm{B}^{(F)}$. We skip all further technical details here (q.v. Section 4.6), but a final caveat seems in place: In general, the last lemma above is not true without the condition that P is a polytope, even if P is still assumed to be convex and compact! Also note that our whole line of argument would be vastly more complicated with cheating strategies instead of cheating situation, what again proves usefulness of the latter concept.

Exploiting decomposability of cheating situations. So far, we have reached two essential insights. On the one hand, by Section 3.1.3 we can find OT-cores, such that for Bob's corresponding input symbols $\bar{y}, \bar{y}'$ and every cheating situation $\eta \in \mathfrak{N}_\mathrm{B}^{(F)}$ with $\eta|_\mathrm{B}^{\mathsf{fake}}(\{\bar{y},\bar{y}'\}) = 1$ we have that $\eta|_\mathrm{B}^{\mathsf{true}}(\{\bar{y},\bar{y}'\}) = 1$ (cf. left diagram in Figure 3.3). On the other hand, by the considerations above we know that a real protocol run with overwhelming probability is either aborted or there exists a cheating situation $\eta \in \mathfrak{N}_\mathrm{B}^{(F)}$, such that $\eta|_\mathrm{B}^{\mathsf{true}}$ is $k^{-\varepsilon}$-close to Bob's actual input distribution and $\eta|_\mathrm{B}^{\mathsf{fake}}$ is $k^{-\varepsilon}$-close to Bob's claimed input distribution. Since otherwise Alice aborts the protocol, the latter also implies that $\eta|_\mathrm{B}^{\mathsf{fake}}$ is $k^{-\varepsilon}$-close to Bob's prescribed input distribution.

Now we want to tie these two things together, but we have the following problem. As discussed right at the start of Section 3.1, the support of Bob's prescribed input distribution must be his complete input alphabet Υ_B. Hence, we must also tolerate in the protocol step Check A that Bob sometimes claims to have used an input symbol that does not belong to the chosen OT-core. Thereby, we only get that $\eta|_\mathrm{B}^{\mathsf{fake}}(y') \leq k^{-\delta}$ for all $y' \in \Upsilon_\mathrm{B}\backslash\{\bar{y},\bar{y}'\}$ with constant $\delta > 0$, rather than $\eta|_\mathrm{B}^{\mathsf{fake}}(\Upsilon_\mathrm{B}\backslash\{\bar{y},\bar{y}'\}) = 0$ (cf. middle diagram in Figure 3.3).

We solve this problem by exploiting the fact that, up to rescaling of Alice's input distribution $\eta|_\mathrm{A}$, the set $\mathfrak{N}_\mathrm{B}^{(F)}$ is the convex hull of a finite spanning set $\{\dot{\eta}_1,\ldots,\dot{\eta}_n\} \subseteq \mathfrak{N}_\mathrm{B}^{(F)}$. Since $k^{-\delta}$ becomes arbitrarily small for increasing security parameter k, but there exists some constant $\vartheta_F > 0$ with $\dot{\eta}_i|_\mathrm{B}^{\mathsf{fake}}(y') \notin (0,\vartheta_F)$ for all $y' \in \Upsilon_\mathrm{B}, i \in \{1,\ldots,n\}$, we can conclude that our initially given cheating situation η consists only in small part of cheating situations $\dot{\eta}_i$ with $\dot{\eta}_i|_\mathrm{B}^{\mathsf{fake}}(\Upsilon_\mathrm{B}\backslash\{\bar{y},\bar{y}'\}) > 0$ (cf. right diagram in Figure 3.3). Thus, we only introduce an additional error of magnitude $O(k^{-\delta})$, if we approximate a cheating Bob's behavior by a cheating situation $\eta \in \mathfrak{N}_\mathrm{B}^{(F)}$ with $\eta|_\mathrm{B}^{\mathsf{fake}}(\{\bar{y},\bar{y}'\}) = 1$. So, after all we can utilize that our chosen OT-core does not allow for non-trivial cheating situations, and we can conclude that Bob has to play honestly up to some polynomially vanishing fraction of his inputs.

Functionality $\mathcal{F}_{\text{SMCD}}^{(G,\varepsilon)}$

Parametrized by a constant $\varepsilon > 0$ and $G := (\Lambda_A, \Lambda_B, \psi)$, where Λ_A and Λ_B are finite alphabets and $\psi \in \text{pmf}(\Lambda_A \times \Lambda_B)$, such that $\psi(\alpha, \Lambda_B) > 0$ for all $\alpha \in \Lambda_A$ and $\psi(\Lambda_A, \beta) > 0$ for all $\beta \in \Lambda_B$. Let k denote the security parameter and let $\Delta := 1 - \varepsilon$.

- Wait for the adversary to send a compound string $t_A \times t_B$, such that $k - k^{1-\varepsilon} \leq |t_A \times t_B| \leq k$ and for all $\alpha \in \Lambda_A$, $\beta \in \Lambda_B$ it holds:
$$|t_A \times t_B|_{(\alpha,\beta)} \;=\; k \cdot \psi(\alpha, \beta) \pm k^{\Delta}$$
Further, if Alice is uncorrupted, it must hold that $t_A \in \Lambda_A^*$ (but not necessarily $t_B \in \Lambda_B^*$). Analogously, if Bob is uncorrupted, it must hold that $t_B \in \Lambda_B^*$ (but not necessarily $t_A \in \Lambda_A^*$).
If such a compound string $t_A \times t_B$ is sent for the first time, resample each $t_A[i]$ and/or $t_B[i]$ as follows:
  - If no party is corrupted, resample the complete tuple $\big(t_A[i], t_B[i]\big)$ according to ψ.
  - If only Alice is corrupted and $t_A[i] \in \Lambda_A$, then resample $t_B[i]$, s.t. $\mathbb{P}\big[t_B[i] = \beta\big] = \frac{\psi(t_A[i],\beta)}{\psi(t_A[i],\Lambda_B)}$.
  - If only Bob is corrupted and $t_B[i] \in \Lambda_B$, then resample $t_A[i]$, s.t. $\mathbb{P}\big[t_A[i] = \alpha\big] = \frac{\psi(\alpha,t_B[i])}{\psi(\Lambda_A,t_B[i])}$.
  - If both parties are corrupted or $\big(t_A[i], t_B[i]\big) \notin \Lambda_A \times \Lambda_B$, then neither resample $t_A[i]$ nor $t_B[i]$.

  Next, record the resulting compound string $t_A \times t_B$. Henceforth, ignore any further $t_A \times t_B$-messages from the adversary.
- Upon receiving a message $(\texttt{Delivery}, \texttt{Alice})$ from the adversary, verify that there is a stored compound string $t_A \times t_B$; else ignore that message. Next, output t_A to Alice and henceforth ignore all messages $(\texttt{Delivery}, \texttt{Alice})$.
- Upon receiving a message $(\texttt{Delivery}, \texttt{Bob})$ from the adversary, verify that there is a stored compound string $t_A \times t_B$; else ignore that message. Next, output t_B to Bob and henceforth ignore all messages $(\texttt{Delivery}, \texttt{Bob})$.
- Upon receiving any message from Alice or Bob, just forward it to the adversary, inclusive the original sender ID.

Figure 3.4: Ideal functionality for correlated data generation by our protocol scheme based on finite randomized 2-party functions. Corrupted parties have full control over the order of their output string, since order is nowhere checked in our protocol scheme. The resampling just ensures that corrupted parties have no information about honest parties' outputs, other than what they learn by their own output. The condition that $\psi(\alpha, \Lambda_B) > 0$ and $\psi(\Lambda_A, \beta) > 0$ for all $\alpha \in \Lambda_A$, $\beta \in \Lambda_B$ is needed to avoid division by zero during the resampling process. The constant Δ is motivated by the Hoeffding Inequality (q.v. Section 3.1.4); w.l.o.g. we always have that $\varepsilon < \frac{1}{2}$ and thus $\Delta > \frac{1}{2}$.

Secure generation of correlated data. Putting things together, we have shown that in the protocol scheme introduced in Section 3.1.1, if instantiated appropriately, even corrupted parties cannot deviate too much from the prescribed input distributions without being caught cheating. Furthermore, the final protocol output consists for the most part of such "almost honestly" generated data (cf. the final discussion of Section 3.1.1), even if a corrupted party chooses a challenge set maliciously in one of the check steps, and/or lies in the final output step about which inputs did not belong to the chosen OT-core. Altogether, our protocol produces some "slightly manipulable correlated data" (SMCD). We want to grasp this by defining an according functionality, which is implemented by our protocol in the UC sense, but first we need to introduce some suitable notation.

Notation. Given a finite string s over some alphabet Ω, let $|s|$ denote the length of s. By $|s|_\alpha$ with $\alpha \in \Omega$ we denote the number of appearances of α in s. By $s[i]$ with $i \in \{1, \ldots, |s|\}$ we denote the i-th element of s. For some given strings s_A and s_B of the same length, we define the *compound string* $s_A \times s_B$, whose i-th element is just the tuple $\big(s_A[i], s_B[i]\big)$.

a)

$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{2}$	
$\frac{1}{6}$	$\frac{1}{3}$		$\frac{1}{2}$

b)

$\frac{1}{12}$	$\frac{1}{6}$	$\frac{1}{4}$	
$\frac{1}{12}$	$\frac{1}{6}$		$\frac{1}{4}$

c)

$\frac{1}{4}$	$\frac{1}{4}$
$\frac{1}{4}$	$\frac{1}{4}$

Figure 3.5: Canonical representation of a 2-party function (a), the resulting correlated data distribution (b), and condensed version of the latter (c).

Now, let us consider the protocol scheme from Section 3.1.1, instantiated as follows.

- The underlying 2-party function $F := (\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Omega_{\mathrm{A}}, \Omega_{\mathrm{B}}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$ is redundancy-free.
- The canonical representation of F has an OT-core that is robust in the sense of Section 3.1.3.
- Alice and Bob each have to use their respective OT-core inputs $\bar{x}, \bar{x}'$ and $\bar{y}, \bar{y}'$ with equal probability, and all other input symbols with some polynomially vanishing probability.
- There exists some constant $\varepsilon > 0$, such that $k - k^{1-\varepsilon}$ elements of the final output strings are generated honestly, even if one party is corrupted (cf. the final discussion in Section 3.1.1).

Given such a setting, our protocol implements the functionality $\mathcal{F}_{\mathrm{SMCD}}^{(G,\varepsilon)}$ defined in Figure 3.4, where $G = (\Lambda_{\mathrm{A}}, \Lambda_{\mathrm{B}}, \psi)$, instantiated as follows (cf. Section 4.8):

$$\Lambda_{\mathrm{A}} = \{\bar{x}, \bar{x}'\} \times \Omega_{\mathrm{A}} \qquad \Lambda_{\mathrm{B}} = \{\bar{y}, \bar{y}'\} \times \Omega_{\mathrm{B}} \qquad \psi\big((x,a),(y,b)\big) = \frac{\phi_{x,y}(a,b)}{\big|\{\bar{x}, \bar{x}'\} \times \{\bar{y}, \bar{y}'\}\big|}$$

3.2 Reduction of OT to correlated data

In Section 3.1 we have seen how to securely generate non-trivially correlated data from any redundancy-free 2-party function that has some OT-core. Now we have to implement OT from such data, i.e. we have to construct an OT protocol based on the functionality $\mathcal{F}_{\mathrm{SMCD}}^{(G,\varepsilon)}$ in Figure 3.4. This protocol construction is only a minor contribution, since in large part the used techniques are just adopted from the standard literature (cf. Section 1.1) and in particular from [CMW05].

Note that in a straightforward manner we can identify $G := (\Lambda_{\mathrm{A}}, \Lambda_{\mathrm{B}}, \psi)$ with a special 2-party function $F := (\{\epsilon\}, \{\epsilon\}, \Lambda_{\mathrm{A}}, \Lambda_{\mathrm{B}}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$ with $\phi_{\epsilon,\epsilon} = \psi$, although the functionality $\mathcal{F}_{\mathrm{SMCD}}^{(G,\varepsilon)}$ works completely different from $\mathcal{F}_{\mathrm{SFE}}^{(F)}$. However, our notions of canonical representations, condensed canonical representations, isomorphism (q.v. Section 2.2) and OT-cores (q.v. Section 2.3) directly carry over (cf. also Figure 3.5). Our notion of redundancy does not apply, since there are no meaningful inputs anymore. In the upcoming sections we always consider the functionality $\mathcal{F}_{\mathrm{SMCD}}^{(G,\varepsilon)}$, where $G = (\Lambda_{\mathrm{A}}, \Lambda_{\mathrm{B}}, \psi)$ and G has some OT-core. W.l.o.g., G is always given in condensed form, meaning that the rows of its canonical representation are pairwise linearly independent and so are the columns.

3.2.1 Refining the correlated data

Removal of unnecessary output symbols. The joint distribution ψ can still be fairly complex, but by the following protocol we can iteratively remove specific symbols from Λ_{A} and also analogously from Λ_{B}, until $|\Lambda_{\mathrm{A}}| = |\Lambda_{\mathrm{B}}| = 2$ and thus w.l.o.g. $\Lambda_{\mathrm{A}} = \Lambda_{\mathrm{B}} = \{0,1\}$. Let $\hat{\alpha}$ denote the symbol to be removed from Λ_{A}. W.l.o.g., G is given in condensed form, i.e. there is no other row in the canonical representation that linearly depends on the $\hat{\alpha}$-row. For our upcoming protocol we will also need that the $\hat{\alpha}$-row in the canonical representation of G is no convex combination of any other rows. And since we do not want to destroy the last OT-core of G by removal of $\hat{\alpha}$, there must exist some OT-core outside of the $\hat{\alpha}$-row. However, as one verifies straightforwardly, $\hat{\alpha}$ can always be chosen this way, if only $|\Lambda_{\mathrm{A}}| > 2$ (remember that we assumed G to be given in condensed form) and G has an OT-core at all. The protocol for removing $\hat{\alpha}$ now just proceeds as follows.

1. Alice announces the index set $I := \{i \in \mathbb{N} \mid t_A[i] = \hat{\alpha}\}$.
2. Bob verifies that $|t_B[I]|_\beta = k \cdot \psi(\hat{\alpha}, \beta) \pm k^\Delta$ for all $\beta \in \Lambda_B$, where $t_B[I]$ denotes the substring of t_B indexed by I; otherwise he aborts the protocol. If there exists any $\beta \in \Lambda_B$, such that $\psi(\Lambda_A \backslash \{\hat{\alpha}\}, \beta) = 0$ and $|t_B[I]|_\beta < |t_B|_\beta$, he also aborts the protocol.
3. Alice and Bob remove the elements indexed by I from t_A and t_B respectively.

Note that Alice cannot lie substantially often, if only the $\hat{\alpha}$-row in the canonical representation is no convex combination of other rows. Thus, we can implement this way $\mathcal{F}_{\text{SMCD}}^{G',\varepsilon'}$ from $\mathcal{F}_{\text{SMCD}}^{G,\varepsilon}$, where $0 < \varepsilon' < \varepsilon$ and G' is obtained from G just by removing $\hat{\alpha}$ from Λ_A and some rescaling of ψ. In particular, we have that $G' = (\Lambda'_A, \Lambda_B, \psi')$, where $\Lambda'_A = \Lambda_A \backslash \{\hat{\alpha}\}$ and $\psi'(\alpha, \beta) = \frac{\psi(\alpha,\beta)}{1-\psi(\hat{\alpha},\Lambda_B)}$ for all $\alpha \in \Lambda'_A, \beta \in \Lambda_B$. Note that this also linearly scales down the security parameter by the factor $1 - \psi(\hat{\alpha}, \Lambda_B)$. Moreover, after removing the $\hat{\alpha}$-row from the canonical representation of G, several columns may become pairwise linearly dependent, what results in a considerably smaller condensed canonical representation of G'. However, as long as the canonical representation of G without the $\hat{\alpha}$-row still contains an OT-core, G' will also have one—full-rank submatrices cannot be completely destroyed by just adding up pairwise linearly dependent columns. So, by iterated removal of single input symbols, we end up with a condensed canonical representation that just *is* an OT-core. Finally, if $\psi(\alpha, \Lambda_B) > \psi(\alpha', \Lambda_B)$ for some $\alpha, \alpha' \in \Lambda_A$, we let Alice analogously remove some α-elements from t_A, so that afterwards $\psi(\alpha, \Lambda_B) = \psi(\alpha', \Lambda_B)$ for all $\alpha, \alpha' \in \Lambda_A$.

This removal of unnecessary output symbols and balancing of Alice's output distribution is UC-secure; the simulator construction and security proof can be sketched as follows. Talking in terms of the UC framework, we are in the $\mathcal{F}_{\text{SMCD}}^{(G,\varepsilon)}$-hybrid model and want to implement the ideal functionality $\mathcal{F}_{\text{SMCD}}^{G',\varepsilon'}$. If no party is corrupted, the simulator basically needs to send a compound string of correct length to the ideal functionality $\mathcal{F}_{\text{SMCD}}^{G',\varepsilon'}$, and he can produce such a string simply by simulating a complete protocol run with honest parties. If Alice is corrupted, basically all she can do is trying to choose the index set I maliciously. Note that $|t_A|_{\Lambda_A} \geq k - |\Lambda_A \times \Lambda_B| \cdot k^\Delta$ by the construction of $\mathcal{F}_{\text{SMCD}}^{(G,\varepsilon)}$, i.e. $t_A[i] \notin \Lambda_A$ for at most $|\Lambda_A \times \Lambda_B| \cdot k^\Delta$ indices i. Further, by the Hoeffding Inequality we have for every $\beta \in \Lambda_B$ that with overwhelming probability $|t_A \times t_B[I]|_{\Lambda_A \times \{\beta\}} = \sum_{\alpha \in \Lambda_A} |t_A[I]|_\alpha \cdot \frac{\psi(\alpha,\beta)}{\psi(\alpha,\Lambda_B)} \pm k^\Delta$, and hence $|t_B[I]|_\beta = \sum_{\alpha \in \Lambda_A} |t_A[I]|_\alpha \cdot \frac{\psi(\alpha,\beta)}{\psi(\alpha,\Lambda_B)} \pm (1 + |\Lambda_A \times \Lambda_B|) k^\Delta$. Since otherwise Bob aborts the protocol, this implies that $\sum_{\alpha \in \Lambda_A} |t_A[I]|_\alpha \cdot \frac{\psi(\alpha,\beta)}{\psi(\alpha,\Lambda_B)} = k \cdot \psi(\hat{\alpha}, \beta) \pm (2 + |\Lambda_A \times \Lambda_B|) k^\Delta$, or in other words:

$$\sum_{\alpha \in \Lambda_A} \frac{|t_A[I]|_\alpha}{k \cdot \psi(\alpha,\Lambda_B)} \cdot \psi(\alpha, \beta) \;=\; \psi(\hat{\alpha}, \beta) \pm (2 + |\Lambda_A \times \Lambda_B|) k^{-\varepsilon}$$

Since by assumption the $\hat{\alpha}$-row in the canonical representation of G is bounded away from the convex hull of all other rows, this eventually yields that Alice must choose the index set I correctly up to some $O(k^{1-\varepsilon})$-error. This is simulatable, since the ideal functionality $\mathcal{F}_{\text{SMCD}}^{G',\varepsilon'}$ even tolerates $k^{1-\varepsilon'}$-errors, and we have chosen $\varepsilon' < \varepsilon$. Finally, a corrupted Bob can just maliciously abort the protocol, which can be simulated trivially.

Balancing of Bob's output distribution. By the method above we have implemented $\mathcal{F}_{\text{SMCD}}^{(G,\varepsilon)}$ with $G = (\Lambda_A, \Lambda_B, \psi)$, such that w.l.o.g. $\Lambda_A = \Lambda_B = \{0, 1\}$ and $\psi(0, \Lambda_B) = \psi(1, \Lambda_B)$. Moreover, w.l.o.g. we have that $\psi(0,0) \cdot \psi(1,1) > \psi(1,0) \cdot \psi(0,1)$. We now further refine the correlated data, such that afterwards the joint distribution is completely balanced in 0 and 1. Therefor, we need to extend Bob's alphabet Λ_B by a special erasure symbol "$\perp$". In particular, we implement something similar to $\mathcal{F}_{\text{SMCD}}^{(G',\varepsilon')}$ with $G' = (\{0,1\}, \{0,1,\perp\}, \psi')$, such that $\psi'(0,\perp) = \psi'(1,\perp) > 0$ and $\psi'(0,0) = \psi'(1,1) > \psi'(1,0) = \psi'(0,1)$. However, in doing so we will halve the security parameter.

1. Alice deletes $||t_A|_0 - |t_B|_1|$ elements from t_A, such that afterwards $|t_A|_0 = |t_B|_1$. She announces the corresponding indices to Bob, who deletes the according elements from t_B, too. If afterwards $|t_B|$ is not an even number or Alice announced more than k^Δ indices, Bob aborts.

2. Alice randomly permutes t_A subject to the condition that afterwards $t_\mathrm{A}[i] \neq t_\mathrm{A}[i+1]$ for all odd indices i. She announces the permutation to Bob, who permutes t_B the same way.
3. Alice and Bob locally generate new strings $t'_\mathrm{A} \in \{0,1\}^*$ and $t'_\mathrm{B} \in \{0,1,\perp\}^*$ of half length as follows:

$$\begin{array}{ll} t'_\mathrm{A}[i] := 0 & \text{if } (t_\mathrm{A}[2i-1], t_\mathrm{A}[2i]) = (0,1) \\ t'_\mathrm{A}[i] := 1 & \text{if } (t_\mathrm{A}[2i-1], t_\mathrm{A}[2i]) = (1,0) \end{array} \qquad \begin{array}{ll} t'_\mathrm{B}[i] := 0 & \text{if } (t_\mathrm{B}[2i-1], t_\mathrm{B}[2i]) = (0,1) \\ t'_\mathrm{B}[i] := 1 & \text{if } (t_\mathrm{B}[2i-1], t_\mathrm{B}[2i]) = (1,0) \\ t'_\mathrm{B}[i] := \perp & \text{if } t_\mathrm{B}[2i-1] = t_\mathrm{B}[2i] \end{array}$$

4. Bob aborts the protocol, if it holds:

$$|t'_\mathrm{B}|_\perp \;>\; \frac{k}{2} \cdot \frac{\psi(0,0)\cdot\psi(1,0) + \psi(0,1)\cdot\psi(1,1)}{\psi(0,\Lambda_\mathrm{B})\cdot\psi(1,\Lambda_\mathrm{B})} + k^\Delta$$

5. Alice outputs t'_A and Bob outputs t'_B.

This way we get:

$$\psi'(0,0) = \frac{\psi(0,0)\cdot\psi(1,1)}{2\cdot\psi(0,\Lambda_\mathrm{B})\cdot\psi(1,\Lambda_\mathrm{B})} \qquad \psi'(0,\perp) = \frac{\psi(0,0)\cdot\psi(1,0) + \psi(0,1)\cdot\psi(1,1)}{2\cdot\psi(0,\Lambda_\mathrm{B})\cdot\psi(1,\Lambda_\mathrm{B})}$$

$$\psi'(0,1) = \frac{\psi(0,1)\cdot\psi(1,0)}{2\cdot\psi(0,\Lambda_\mathrm{B})\cdot\psi(1,\Lambda_\mathrm{B})} \qquad \psi'(1,\perp) = \frac{\psi(1,0)\cdot\psi(0,0) + \psi(1,1)\cdot\psi(0,1)}{2\cdot\psi(0,\Lambda_\mathrm{B})\cdot\psi(1,\Lambda_\mathrm{B})}$$

$$\psi'(1,0) = \frac{\psi(1,0)\cdot\psi(0,1)}{2\cdot\psi(0,\Lambda_\mathrm{B})\cdot\psi(1,\Lambda_\mathrm{B})}$$

$$\psi'(1,1) = \frac{\psi(1,1)\cdot\psi(0,0)}{2\cdot\psi(0,\Lambda_\mathrm{B})\cdot\psi(1,\Lambda_\mathrm{B})}$$

Note that Bob cannot cheat at all, but he must prevent Alice from maliciously choosing a permutation that yields $t_\mathrm{A}[i] = t_\mathrm{A}[i+1]$ for substantially many odd indices i in the first step. This is what the fourth protocol step is needed for. According to the Hoeffding Inequality, $|t'_\mathrm{B}|_\perp$ is raised asymptotically by $\Omega(\bar{n} - k^\Delta)$, where $\bar{n}$ denotes the number of odd indices i with $t_\mathrm{A}[i] = t_\mathrm{A}[i+1]$ after the permutation. Thus, a corrupted Alice is caught cheating with overwhelming probability, if $\bar{n} \notin O(k^\Delta)$. Moreover, a corrupted Bob has no control over the order of the final output strings any more, due to Alice's random permutation in the first protocol step. Putting things together, there exist some constants $\varepsilon', \varepsilon'', \tilde{\varepsilon}, \tilde{\nu} \in (0,1)$, particularly $\tilde{\nu} = \psi'(\{0,1\},\perp)$ and $\tilde{\varepsilon} = \frac{\psi'(0,1)+\psi'(1,0)}{1-\tilde{\nu}}$, such that we have now the following situation with κ denoting the new security parameter.

- Alice's output is a uniformly random string $t'_\mathrm{A} \in \{0,1\}^\kappa$.
- Bob's output $t'_\mathrm{B} \in \{0,1,\perp\}^\kappa$ is randomly generated according to the following probabilities:

$$\begin{aligned} \mathbb{P}\left[t'_\mathrm{B}[i] = t'_\mathrm{A}[i]\right] &= (1-\tilde{\nu})\cdot(1-\tilde{\varepsilon}) \\ \mathbb{P}\left[t'_\mathrm{B}[i] = \neg t'_\mathrm{A}[i]\right] &= (1-\tilde{\nu})\cdot\tilde{\varepsilon} \\ \mathbb{P}\left[t'_\mathrm{B}[i] = \perp\right] &= \tilde{\nu} \end{aligned}$$

- If Alice is corrupted, she may choose her output $t'_\mathrm{A} \in \{0,1\}^\kappa$ arbitrarily and afterwards learn some additional information about up to $\kappa^{1-\varepsilon'}$ arbitrarily chosen elements of t'_B.
- For up to $\kappa^{1-\varepsilon''}$ random indices $i \in \{1,\ldots,\kappa\}$, a corrupted Bob my learn some additional information about $t'_\mathrm{A}[i]$.

Note that $\tilde{\varepsilon} < \frac{1}{2}$, since $\psi'(0,0) = \psi'(1,1) > \psi'(1,0) = \psi'(0,1)$ by construction. The implemented functionality is different from $\mathcal{F}_\mathrm{SMCD}^{(G',\varepsilon')}$, insofar as a corrupted Bob has no longer control over his output order or for which indices i he gets additional information about $t'_\mathrm{A}[i]$, but a corrupted Alice can now arbitrarily choose her output string t'_A provided that $|t'_\mathrm{A}|_{\{0,1\}} \geq \kappa - \kappa^{1-\varepsilon'}$. Nonetheless, we still have a UC-secure implementation of this modified version of $\mathcal{F}_\mathrm{SMCD}^{(G',\varepsilon')}$. The respective simulator construction and security proof are pretty similar to those for our protocol above for removal of unnecessary output symbols.

3.2.2 Building OT from the refined correlated data

Let $\varepsilon', \varepsilon'', \tilde{\varepsilon}, \tilde{\nu}, \kappa, t'_{\mathrm{A}}, t'_{\mathrm{B}}$ as above. If Alice is honest, we can easily implement from this a non-trivial binary symmetric erasure channel (BSEC) that allows Alice to send κ bits and then shuts down: To send the i-th bit, say $m[i]$, Alice just has to announce $\tilde{m}[i] := m[i] \oplus t'_{\mathrm{A}}[i]$ to Bob. Bob then can recover a noisy version $m'[i]$ of $m[i]$ by computing $m'(i) = \tilde{m}[i] \oplus t'_{\mathrm{B}}[i]$ with the convention that $0 \oplus \perp = 1 \oplus \perp = \perp$. Obviously, the implemented BSEC has the following properties:

- If Bob is honest, the erasure probability is $\tilde{\nu}$.
- If Bob is corrupted, the erasure probability is still lower bounded by $\tilde{\nu} - \kappa^{-\varepsilon''}$.
- If Bob is honest, the error probability is $(1 - \tilde{\nu}) \cdot \tilde{\varepsilon}$.
- If Bob is corrupted, the error probability is still lower bounded by $(1 - \tilde{\nu}) \cdot \tilde{\varepsilon} - \kappa^{-\varepsilon''}$.

I.e., the differences between the channel characteristics for an uncorrupted and a corrupted Bob are polynomially vanishing in the security parameter. Especially, the channel parameters for a corrupted Bob do converge to the corresponding parameters of the honest case. This is good enough, so that our BSEC can be transformed into OT statistically secure against a corrupted receiver party Bob just by applying one of the protocols from the literature [CMW05, Wul09, IKO$^+$11]. Note that although only [IKO$^+$11] is explicitly stated in the UC framework, the security proofs of the other approaches can also be turned into UC proofs rather simply.

However, we still have to take care of a corrupted Alice, who can additionally learn $t'_{\mathrm{B}}[i]$ for up to $\kappa^{1-\varepsilon'}$ arbitrarily chosen indices i. We deal with this as follows. Instead of implementing a single BSEC that can be used κ times, we implement $\ell := \lfloor \kappa^{1-\varepsilon'} + 1 \rfloor$ BSECs that each can be used $\lambda := \lfloor \kappa / \ell \rfloor$ times. We just use the first λ elements of t'_{A} and t'_{B} for the first BSEC, the next λ elements of t'_{A} and t'_{B} for the second BSEC, and so on. Analogously to above, this gives us ℓ OTs with polynomially downscaled security parameter λ, each of which is statistically UC-secure against a corrupted receiver Bob. But now, since a corrupted Alice can cheat at most $\kappa^{1-\varepsilon'}$ times, at last one of these OTs is also statistically UC-secure against Alice. Finally, we can use a simple standard combiner to achieve a fully (i.e. against both parties) UC-secure OT instance:

0. Let (b_0, b_1) denote Alice's respective sender input and let c denote Bob's choice bit.
1. Alice chooses two ℓ-bit strings $\hat{b}_0, \hat{b}_1 \in \{0,1\}^\ell$ uniformly at random, subject to the condition that $\bigoplus_{i=1}^{\ell} \hat{b}_0[i] = b_0$ and $\hat{b}_0[i] \oplus \hat{b}_1[i] = b_0 \oplus b_1$ for all $i \in \{1, \ldots, \ell\}$.
   Bob chooses $\hat{c} \in \{0,1\}^\ell$ uniformly at random, subject to the condition that $\bigoplus_{i=1}^{\ell} \hat{c}[i] = c$.
2. For each $i \in \{1, \ldots, \ell\}$, Alice and Bob run OT with sender input $(\hat{b}_0[i], \hat{b}_1[i])$ from Alice and choice bit $\hat{c}[i]$ from Bob, such that all ℓ OT instances are secure against Bob and at least one instance is secure against Alice.
3. Bob computes and outputs $b_c = \bigoplus_{i=1}^{\ell} \hat{b}_{\hat{c}[i]}[i]$.

It is not hard to verify that this protocol is correct, hides c from Alice, and Bob may learn at most one of the bit values b_0, b_1. Even if Alice maliciously chooses $(\hat{b}_0, \hat{b}_1)$ such that $\hat{b}_0[i] \oplus \hat{b}_1[i]$ is not the same for all $i \in \{1, \ldots, \ell\}$, this means no security violation: It only randomizes Bob's final output, which she could as well achieve by choosing her protocol input (b_0, b_1) just uniformly at random in the first place.

This whole construction can easily be proven UC-secure, since by UC-security of the ℓ OT subprotocols in step 2 of our combiner even for corrupted parties the corresponding inputs $(\hat{b}_0[i], \hat{b}_1[i])$ and $\hat{c}[i]$ are always well-defined. We only have to take into account that a corrupted Alice may additionally learn up to $\ell - 1$ bits of $\hat{c}$. However, this is just pure randomness, uncorrelated with everything else. The general idea of how the simulation in the ideal model works can be described as follows.

- If Alice is corrupted, the simulator lets her run the protocol with a simulated instance of Bob, whose choice bit c is just uniform randomness. After step 2 the simulator can easily extract $b_c = \bigoplus_{i=1}^{\ell} \hat{b}_{\hat{c}[i]}[i]$, compute $b_{\neg c} = b_c \oplus \hat{b}_o[j] \oplus \hat{b}_1[j]$ with j corresponding to an OT instance that were secure against Alice, and finally send (b_0, b_1) to the ideal functionality.

- If Bob is corrupted, the simulator lets him run the protocol with a simulated instance of Alice, whose sender input (b_0, b_1) is just uniform randomness. At the beginning of the final iteration of step 2 the simulator can easily extract $c = \bigoplus_{i=1}^{\ell} \hat{c}[i]$, send c to the ideal functionality, and thus learn b_c. Then, if $b_c \neq \bigoplus_{i=1}^{\ell} \hat{b}_{\hat{c}[i]}[i]$, he just has to flip the bit values of $\hat{b}_0[\ell]$ and $\hat{b}_1[\ell]$ in the simulated Alice's memory before he resumes simulating the rest of step 2. This is perfectly indistinguishable from a real protocol run, unless a corrupted Bob can gather some information about $\hat{b}_0[i] \oplus \hat{b}_1[i]$ for any $i \in \{1, \ldots, \ell\}$. However, the latter is ruled out by security against Bob of the underlying ℓ OT instances.

Once again, we omit the fully detailed UC proof, since it does not contain any additional technical insights. This concludes our more informal exposition of how one can prove the Classification Theorem from Section 2.3.

4 Formal basis

In this chapter, we formally prove that secure generation of correlated data (in the sense of Figure 3.4) can be implemented from any redundancy-free 2-party function $F \in \mathfrak{F}_{\mathsf{fin}}$ that has some OT-core. This is our main technical contribution, since OT can be reduced to such correlated data by rather standard techniques (q.v. Section 3.2).

4.1 Basic notions and notations

We start off with a collective (re)statement of all definitions and notations that are used throughout the rest of this part of the thesis.

Notation 1 (Finite sums of function values). Given a set T with finite subset $S \subseteq T$ and some mapping $g : T \to \mathbb{R}$, we set $g(S) := \sum_{\omega \in S} g(\omega)$. For functions with more arguments we use the canonical extension of this notation, e.g. $h(a, B, C, d) := \sum_{\beta \in B,\, \gamma \in C} h(a, \beta, \gamma, d)$.

Notation 2 (Spaces of probability mass functions). Given some finite alphabet Ω, we denote the set of all probability mass functions over Ω by $\mathrm{pmf}(\Omega)$, i.e. $\mathrm{pmf}(\Omega) = \{\rho : \Omega \to \mathbb{R}_{\geq 0} \mid \rho(\Omega) = 1\}$.

Notation 3 (Finite randomized 2-party functions). Let $\mathfrak{F}_{\mathsf{fin}}$ denote the set of all quintuples $(\Upsilon_\mathrm{A}, \Upsilon_\mathrm{B}, \Omega_\mathrm{A}, \Omega_\mathrm{B}, \phi)$, where $\Upsilon_\mathrm{A}, \Upsilon_\mathrm{B}, \Omega_\mathrm{A}, \Omega_\mathrm{B}$ are non-empty finite alphabets and $\phi = \{\phi_{x,y}\}_{x \in \Upsilon_\mathrm{A}, y \in \Upsilon_\mathrm{B}}$ is a family of probability mass functions over $\Omega_\mathrm{A} \times \Omega_\mathrm{B}$, i.e. $\phi \subseteq \mathrm{pmf}(\Omega_\mathrm{A} \times \Omega_\mathrm{B})$.

Definition 4 (Redundancy)**.** Let $F := (\Upsilon_\mathrm{A}, \Upsilon_\mathrm{B}, \Omega_\mathrm{A}, \Omega_\mathrm{B}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$. An input symbol $y' \in \Upsilon_\mathrm{B}$ is called *redundant*, if there exist some $\tau \in \mathrm{pmf}(\Upsilon_\mathrm{B})$ and a family of probability mass functions $\{\lambda_{y,b}\}_{y \in \Upsilon_\mathrm{B}, b \in \Omega_\mathrm{B}} \subseteq \mathrm{pmf}(\Omega_\mathrm{B})$, such that $\tau(y') = 0$ and for all $x \in \Upsilon_\mathrm{A}$, $a \in \Omega_\mathrm{A}$, $b' \in \Omega_\mathrm{B}$ it holds:

$$\phi_{x,y'}(a, b') = \sum\nolimits_{y \in \Upsilon_\mathrm{B},\, b \in \Omega_\mathrm{B}} \tau(y) \cdot \phi_{x,y}(a, b) \cdot \lambda_{y,b}(b')$$

For input symbols $x \in \Upsilon_\mathrm{A}$ redundancy is defined analogously. If neither Υ_A nor Υ_B contains any redundant input symbols, F is called *redundancy-free*.

Definition 5 (Cheating situations)**.** For $F := (\Upsilon_\mathrm{A}, \Upsilon_\mathrm{B}, \Omega_\mathrm{A}, \Omega_\mathrm{B}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$ let $\mathfrak{N}_\mathrm{B}^{(F)}$ denote the set of all probability mass functions $\eta \in \mathrm{pmf}\big((\Upsilon_\mathrm{A} \times \Omega_\mathrm{A}) \times (\Upsilon_\mathrm{B} \times \Omega_\mathrm{B})^2\big)$ that meet the following conditions.

1. For all $x \in \Upsilon_\mathrm{A}$ it holds that $\eta|_\mathrm{A}(x) := \eta\big((x, \Omega_\mathrm{A}), (\Upsilon_\mathrm{B}, \Omega_\mathrm{B}), (\Upsilon_\mathrm{B}, \Omega_\mathrm{B})\big) > 0$.

2. For all $x \in \Upsilon_\mathrm{A}$, $a \in \Omega_\mathrm{A}$, $y \in \Upsilon_\mathrm{B}$, $b \in \Omega_\mathrm{B}$, with $\eta|_\mathrm{B}^{\mathsf{true}}(y) := \eta\big((\Upsilon_\mathrm{A}, \Omega_\mathrm{A}), (y, \Omega_\mathrm{B}), (\Upsilon_\mathrm{B}, \Omega_\mathrm{B})\big)$, it holds:
$$\eta\big((x, a), (y, b), (\Upsilon_\mathrm{B}, \Omega_\mathrm{B})\big) = \eta|_\mathrm{A}(x) \cdot \eta|_\mathrm{B}^{\mathsf{true}}(y) \cdot \phi_{x,y}(a, b)$$

3. For all $x \in \Upsilon_\mathrm{A}$, $a \in \Omega_\mathrm{A}$, $y' \in \Upsilon_\mathrm{B}$, $b' \in \Omega_\mathrm{B}$, with $\eta|_\mathrm{B}^{\mathsf{fake}}(y') := \eta\big((\Upsilon_\mathrm{A}, \Omega_\mathrm{A}), (y', \Omega_\mathrm{B}), (\Upsilon_\mathrm{B}, \Omega_\mathrm{B})\big)$, it holds:
$$\eta\big((x, a), (\Upsilon_\mathrm{B}, \Omega_\mathrm{B}), (y', b')\big) = \eta|_\mathrm{A}(x) \cdot \eta|_\mathrm{B}^{\mathsf{fake}}(y') \cdot \phi_{x,y'}(a, b')$$

4. For all $x \in \Upsilon_\mathrm{A}$, $a \in \Omega_\mathrm{A}$, $y, y' \in \Upsilon_\mathrm{B}$, $b, b' \in \Omega_\mathrm{B}$ with $\big((\Upsilon_\mathrm{A}, \Omega_\mathrm{A}), (y, b), (\Upsilon_\mathrm{B}, \Omega_\mathrm{B})\big) > 0$ it holds:
$$\eta\big((x, a), (y, b), (y', b')\big) = \eta\big((\Upsilon_\mathrm{A}, \Omega_\mathrm{A}), (y, b), (y', b')\big) \cdot \frac{\eta\big((x, a), (y, b), (\Upsilon_\mathrm{B}, \Omega_\mathrm{B})\big)}{\eta\big((\Upsilon_\mathrm{A}, \Omega_\mathrm{A}), (y, b), (\Upsilon_\mathrm{B}, \Omega_\mathrm{B})\big)}$$

The mappings $\eta \in \mathfrak{N}_{\mathrm{B}}^{(F)}$ are called *Bob's cheating situations for F*. The set $\mathfrak{N}_{\mathrm{A}}^{(F)}$ of *Alice's cheating situations for F* is defined analogously.

Definition 6 (Special cheating situations)**.** Let $F := (\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Omega_{\mathrm{A}}, \Omega_{\mathrm{B}}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$. We define the following subsets of $\mathfrak{N}_{\mathrm{B}}^{(F)}$.

Normalized cheating situations: A cheating situation $\eta \in \mathfrak{N}_{\mathrm{B}}^{(F)}$ is called *normalized*, if $\eta|_{\mathrm{A}}(x) = \frac{1}{|\Upsilon_{\mathrm{A}}|}$ for all $x \in \Upsilon_{\mathrm{A}}$.

Trivial cheating situations: A cheating situation $\eta \in \mathfrak{N}_{\mathrm{B}}^{(F)}$ is called *trivial*, if for all $y, y' \in \Upsilon_{\mathrm{B}}$ and $b, b' \in \Omega_{\mathrm{B}}$ the following implication holds true:

$$(y, b) \neq (y', b') \quad \Rightarrow \quad \eta((\Upsilon_{\mathrm{A}}, \Omega_{\mathrm{A}}), (y, b), (y', b')) = 0$$

Direct cheating situations: A cheating situation $\eta \in \mathfrak{N}_{\mathrm{B}}^{(F)}$ is called *direct*, if for each $(y, b) \in \Upsilon_{\mathrm{B}} \times \Omega_{\mathrm{B}}$ at least one of the following two equalities holds true:

$$\eta((\Upsilon_{\mathrm{A}}, \Omega_{\mathrm{A}}), (y, b), (\Upsilon_{\mathrm{B}}, \Omega_{\mathrm{B}})) = \eta((\Upsilon_{\mathrm{A}}, \Omega_{\mathrm{A}}), (y, b), (y, b))$$
$$\eta((\Upsilon_{\mathrm{A}}, \Omega_{\mathrm{A}}), (\Upsilon_{\mathrm{B}}, \Omega_{\mathrm{B}}), (y, b)) = \eta((\Upsilon_{\mathrm{A}}, \Omega_{\mathrm{A}}), (y, b), (y, b))$$

Straight cheating situations: A cheating situation $\eta \in \mathfrak{N}_{\mathrm{B}}^{(F)}$ is called *straight*, if for each $y \in \Upsilon_{\mathrm{B}}$ at least one of the following two equalities holds true:

$$\hat{\eta}|_{\mathrm{B}}^{\mathsf{true}}(y) = 0 \qquad \text{or} \qquad \hat{\eta}|_{\mathrm{B}}^{\mathsf{fake}}(y) = 0$$

Definition 7 (Cheating characteristics)**.** For $F := (\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Omega_{\mathrm{A}}, \Omega_{\mathrm{B}}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$ let $\mathfrak{X}_{\mathrm{B}}^{(F)}$ denote the set of all mappings $\xi : \Upsilon_{\mathrm{B}} \to \mathbb{R}$ for that exist some cheating situation $\eta \in \mathfrak{N}_{\mathrm{B}}^{(F)}$ and some $\gamma \in \mathbb{R}_{>0}$, such that for all $y \in \Upsilon_{\mathrm{B}}$ it holds:

$$\gamma \cdot \xi(y) = \eta|_{\mathrm{B}}^{\mathsf{fake}}(y) - \eta|_{\mathrm{B}}^{\mathsf{true}}(y)$$

4.2 Linear properties of cheating situations

In this section we show that cheating situations can be considered independent from the honest party's input distribution, since they can be canonically rescaled (Lemma 8 and Corollary 9). Further, we show how the algebraic structures $\mathfrak{N}_{\mathrm{B}}^{(F)}$ and $\mathfrak{X}_{\mathrm{B}}^{(F)}$ allow for basic composition and/or decomposition of mixed strategies (Lemma 10 and Corollary 11).

Lemma 8 (Rescalability of cheating situations)**.** *Let $F := (\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Omega_{\mathrm{A}}, \Omega_{\mathrm{B}}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$ and $\eta \in \mathfrak{N}_{\mathrm{B}}^{(F)}$. Further, let $\sigma : \Upsilon_{\mathrm{A}} \to \mathbb{R}_{>0}$, such that $\sum_{x \in \Upsilon_{\mathrm{A}}} \sigma(x) \cdot \eta|_{\mathrm{A}}(x) = 1$. Then, the following mapping is a cheating situation for F:*

$$\tilde{\eta} : (\Upsilon_{\mathrm{A}} \times \Omega_{\mathrm{A}}) \times (\Upsilon_{\mathrm{B}} \times \Omega_{\mathrm{B}})^2 \to \mathbb{R}_{\geq 0},\ ((x, a), (y, b), (y', b')) \mapsto \sigma(x) \cdot \eta((x, a), (y, b), (y', b'))$$

Proof. We just have to check the conditions of Definition 5.

0. First note that $\tilde{\eta} \in \mathfrak{N}_{\mathrm{B}}^{(F)}$, since $\mathrm{Image}(\tilde{\eta}) \subseteq \mathbb{R}_{\geq 0}$ and by construction we have:

$$\tilde{\eta}((\Upsilon_{\mathrm{A}}, \Omega_{\mathrm{A}}), (\Upsilon_{\mathrm{B}}, \Omega_{\mathrm{B}}), (\Upsilon_{\mathrm{B}}, \Omega_{\mathrm{B}})) = \sum\nolimits_{x \in \Upsilon_{\mathrm{A}}} \sigma(x) \cdot \eta((x, \Omega_{\mathrm{A}}), (\Upsilon_{\mathrm{B}}, \Omega_{\mathrm{B}}), (\Upsilon_{\mathrm{B}}, \Omega_{\mathrm{B}})) = 1$$

1. For all $x \in \Upsilon_{\mathrm{A}}$ it holds that $\tilde{\eta}|_{\mathrm{A}}(x) = \sigma(x) \cdot \eta|_{\mathrm{A}}(x) > 0$.

2. For all $x \in \Upsilon_{\mathrm{A}}$, $a \in \Omega_{\mathrm{A}}$, $y \in \Upsilon_{\mathrm{B}}$, $b \in \Omega_{\mathrm{B}}$ it holds:

$$\tilde{\eta}\big((x,a),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big) = \sigma(x)\cdot\eta|_{\mathrm{A}}(x)\cdot\eta|_{\mathrm{B}}^{\mathsf{true}}(y)\cdot\phi_{x,y}(a,b)$$

By taking the sum over x, a, b it follows that $\tilde{\eta}|_{\mathrm{B}}^{\mathsf{true}} = \eta|_{\mathrm{B}}^{\mathsf{true}}$. This yields:

$$\tilde{\eta}\big((x,a),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big) = \underbrace{\sigma(x)\cdot\eta|_{\mathrm{A}}(x)}_{\tilde{\eta}|_{\mathrm{A}}(x)}\cdot\underbrace{\eta|_{\mathrm{B}}^{\mathsf{true}}(y)}_{\tilde{\eta}|_{\mathrm{B}}^{\mathsf{true}}(y)}\cdot\phi_{x,y}(a,b)$$

3. For all $x \in \Upsilon_{\mathrm{A}}$, $a \in \Omega_{\mathrm{A}}$, $y' \in \Upsilon_{\mathrm{B}}$, $b' \in \Omega_{\mathrm{B}}$ it holds:

$$\tilde{\eta}\big((x,a),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}}),(y',b')\big) = \sigma(x)\cdot\eta|_{\mathrm{A}}(x)\cdot\eta|_{\mathrm{B}}^{\mathsf{fake}}(y)\cdot\phi_{x,y}(a,b)$$

By taking the sum over x, a, b' it follows that $\tilde{\eta}|_{\mathrm{B}}^{\mathsf{fake}} = \eta|_{\mathrm{B}}^{\mathsf{fake}}$. This yields:

$$\tilde{\eta}\big((x,a),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}}),(y',b')\big) = \underbrace{\sigma(x)\cdot\eta|_{\mathrm{A}}(x)}_{\tilde{\eta}|_{\mathrm{A}}(x)}\cdot\underbrace{\eta|_{\mathrm{B}}^{\mathsf{fake}}(y)}_{\tilde{\eta}|_{\mathrm{B}}^{\mathsf{fake}}(y)}\cdot\phi_{x,y}(a,b)$$

4. For all $x \in \Upsilon_{\mathrm{A}}$, $a \in \Omega_{\mathrm{A}}$, $y, y' \in \Upsilon_{\mathrm{B}}$, $b, b' \in \Omega_{\mathrm{B}}$ with $\eta\big((x,a),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big) > 0$ it holds:

$$\frac{\eta\big((x,a),(y,b),(y',b')\big)}{\eta\big((x,a),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big)} = \frac{\eta\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(y,b),(y',b')\big)}{\eta\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big)}$$

Thereby for all $y, y' \in \Upsilon_{\mathrm{B}}$, $b, b' \in \Omega_{\mathrm{B}}$ we can conclude:

$$\begin{aligned}
&\eta\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(y,b),(y',b')\big)\cdot\textstyle\sum_{x\in\Upsilon_{\mathrm{A}},\,a\in\Omega_{\mathrm{A}}}\sigma(x)\cdot\eta\big((x,a),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big)\\
&=\textstyle\sum_{x\in\Upsilon_{\mathrm{A}},\,a\in\Omega_{\mathrm{A}}}\sigma(x)\cdot\eta\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(y,b),(y',b')\big)\cdot\eta\big((x,a),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big)\\
&=\textstyle\sum_{x\in\Upsilon_{\mathrm{A}},\,a\in\Omega_{\mathrm{A}}}\sigma(x)\cdot\eta\big((x,a),(y,b),(y',b')\big)\cdot\eta\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big)\\
&=\eta\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big)\cdot\textstyle\sum_{x\in\Upsilon_{\mathrm{A}},\,a\in\Omega_{\mathrm{A}}}\sigma(x)\cdot\eta\big((x,a),(y,b),(y',b')\big)
\end{aligned}$$

In other words, for all $y, y' \in \Upsilon_{\mathrm{B}}$, $b, b' \in \Omega_{\mathrm{B}}$ with $\eta\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big) > 0$ it holds:

$$\frac{\eta\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(y,b),(y',b')\big)}{\eta\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big)} = \frac{\sum_{x\in\Upsilon_{\mathrm{A}},\,a\in\Omega_{\mathrm{A}}}\sigma(x)\cdot\eta\big((x,a),(y,b),(y',b')\big)}{\sum_{x\in\Upsilon_{\mathrm{A}},\,a\in\Omega_{\mathrm{A}}}\sigma(x)\cdot\eta\big((x,a),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big)}$$

For all $x \in \Upsilon_{\mathrm{A}}$, $a \in \Omega_{\mathrm{A}}$, $y, y' \in \Upsilon_{\mathrm{B}}$, $b, b' \in \Omega_{\mathrm{B}}$ with $\tilde{\eta}\big((x,a),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big) > 0$ now follows:

$$\begin{aligned}
\frac{\tilde{\eta}\big((x,a),(y,b),(y',b')\big)}{\tilde{\eta}\big((x,a),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big)} &= \frac{\sigma(x)\cdot\eta\big((x,a),(y,b),(y',b')\big)}{\sigma(x)\cdot\eta\big((x,a),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big)}\\
&= \frac{\eta\big((x,a),(y,b),(y',b')\big)}{\eta\big((x,a),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big)}\\
&= \frac{\eta\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(y,b),(y',b')\big)}{\eta\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big)}\\
&= \frac{\sum_{x'\in\Upsilon_{\mathrm{A}},\,a'\in\Omega_{\mathrm{A}}}\sigma(x')\cdot\eta\big((x',a'),(y,b),(y',b')\big)}{\sum_{x'\in\Upsilon_{\mathrm{A}},\,a'\in\Omega_{\mathrm{A}}}\sigma(x')\cdot\eta\big((x',a'),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big)}\\
&= \frac{\tilde{\eta}\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(y,b),(y',b')\big)}{\tilde{\eta}\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big)}
\end{aligned}$$

□

Corollary 9 (Normalizability of cheating situations). *Let* $F := (\Upsilon_A, \Upsilon_B, \Omega_A, \Omega_B, \phi) \in \mathfrak{F}_{\mathsf{fin}}$ *and let* $\eta \in \mathfrak{N}_B^{(F)}$. *Then there exists a unique normalized cheating situation* $\tilde{\eta} \in \mathfrak{N}_B^{(F)}$, *such that for all* $x \in \Upsilon_A$, $a \in \Omega_A$, $y, y' \in \Upsilon_B$, $b, b' \in \Omega_B$ *it holds:*

$$\frac{\eta((x,a),(y,b),(y',b'))}{\eta|_A(x)} = \frac{\tilde{\eta}((x,a),(y,b),(y',b'))}{\tilde{\eta}|_A(x)}$$

Proof. This directly follows by Lemma 8, instantiated as follows:

$$\sigma : \Upsilon_A \to \mathbb{R}_{>0},\ x \mapsto \frac{1}{|\Upsilon_A| \cdot \eta|_A(x)}$$ □

Lemma 10 (Convex combinability of cheating situations). *Let* $F := (\Upsilon_A, \Upsilon_B, \Omega_A, \Omega_B, \phi) \in \mathfrak{F}_{\mathsf{fin}}$ *and* $\upsilon \in \mathrm{pmf}(\Upsilon_A)$, *such that* $\upsilon(x) > 0$ *for all* $x \in \Upsilon_A$. *Then the set of all cheating situations* $\eta \in \mathfrak{N}_B(F)$ *with* $\eta|_A = \upsilon$ *is the convex hull of some finite set of vertices in the affine space* $\mathbb{R}^{(\Upsilon_A \times \Omega_A) \times (\Upsilon_B \times \Omega_B)^2}$. *In particular, for all* $\eta, \eta' \in \mathfrak{N}_B^{(F)}$ *with* $\eta|_A = \eta'|_A = \upsilon$ *and each* $s \in \mathbb{R}$ *the mapping* $\tilde{\eta} := s \cdot \eta + (1-s) \cdot \eta'$ *is a normalized cheating situation for* F, *if only* $\mathrm{Image}(\tilde{\eta}) \subseteq \mathbb{R}_{\geq 0}$.

Proof. It suffices to give a proof for the case that υ is the uniform distribution, i.e. $\upsilon(x) = \frac{1}{|\Upsilon_A|}$ for all $x \in \Upsilon_A$ and thus all considered cheating situations are normalize (q.v. Definition 6). Everything else then follows straightforwardly by Corollary 9.

We just have to adapt the four conditions of Definition 5 to normalized cheating situations. As one verifies quite straightforwardly, the set of all normalized cheating situations for F is the set of all $\eta \in \mathrm{pmf}\big((\Upsilon_A \times \Omega_A) \times (\Upsilon_B \times \Omega_B)^2\big)$ that meet the following conditions.

1. For all $x \in \Upsilon_A$ it holds that $\eta((x, \Omega_A), (\Upsilon_B, \Omega_B), (\Upsilon_B, \Omega_B)) = \frac{1}{|\Upsilon_A|}$.
2. For all $x \in \Upsilon_A$, $a \in \Omega_A$, $y \in \Upsilon_B$, $b \in \Omega_B$ it holds:
$$\eta((x,a),(y,b),(\Upsilon_B,\Omega_B)) = \tfrac{1}{|\Upsilon_A|} \cdot \eta((\Upsilon_A,\Omega_A),(y,\Omega_B),(\Upsilon_B,\Omega_B)) \cdot \phi_{x,y}(a,b)$$
3. For all $x \in \Upsilon_A$, $a \in \Omega_A$, $y' \in \Upsilon_B$, $b' \in \Omega_B$ it holds:
$$\eta((x,a),(\Upsilon_B,\Omega_B),(y',b')) = \tfrac{1}{|\Upsilon_A|} \cdot \eta((\Upsilon_A,\Omega_A),(\Upsilon_B,\Omega_B),(y',\Omega_B)) \cdot \phi_{x,y'}(a,b')$$
4. For all $x \in \Upsilon_A$, $a \in \Omega_A$, $y, y' \in \Upsilon_B$, $b, b' \in \Omega_B$ with $\phi_{\Upsilon_A,y}(\Omega_A, b) > 0$ it holds:
$$\eta((x,a),(y,b),(y',b')) = \eta((\Upsilon_A,\Omega_A),(y,b),(y',b')) \cdot \frac{\phi_{x,y}(a,b)}{\phi_{\Upsilon_A,y}(\Omega_A,b)}$$

Since all these conditions are linear, they define a convex polytope in $\mathbb{R}^{(\Upsilon_A \times \Omega_A) \times (\Upsilon_B \times \Omega_B)^2}$. Note that this polytope is a subset of the bounded set $\mathrm{pmf}\big((\Upsilon_A \times \Omega_A) \times (\Upsilon_B \times \Omega_B)^2\big)$ and hence also is bounded. Further, as the polytope is described by a finite number of linear constraints, it is the convex hull of a finite set of vertices. Finally, the only inequation that normalized cheating situations must fulfill, is that they have non-negative image space. Thus, for all normalized $\eta, \eta' \in \mathfrak{N}_B^{(F)}$ and each $s \in \mathbb{R}$ the mapping $\tilde{\eta} := s \cdot \eta + (1-s) \cdot \eta'$ is a normalized cheating situation for F, if only $\mathrm{Image}(\tilde{\eta}) \subseteq \mathbb{R}_{\geq 0}$. □

Corollary 11 (Positive linearity of cheating characteristics). *Let any* $F \in \mathfrak{F}_{\mathsf{fin}}$ *and* $\xi, \xi' \in \mathfrak{X}_B^{(F)}$, $\gamma, \gamma' \in \mathbb{R}_{>0}$ *be given. Then it holds that* $\gamma \cdot \xi + \gamma' \cdot \xi' \in \mathfrak{X}_B^{(F)}$.

Proof. This directly follows by Definition 7 and the combination of Corollary 9 and Lemma 10. □

4.3 Cheating situations for redundant input symbols

We expose now the inherent structure of cheating strategies by decomposing them into more easily understandable parts. This decomposition consists of two steps.

1. Every cheating situation $\eta \in \mathfrak{N}_{\mathrm{B}}^{(F)}$ is *equivalent* to a *direct* cheating situation $\tilde{\eta} \in \mathfrak{N}_{\mathrm{B}}^{(F)}$, in the sense that $\eta|_{\mathrm{B}}^{\mathsf{true}}(y) = \tilde{\eta}|_{\mathrm{B}}^{\mathsf{true}}(y)$ and $\eta|_{\mathrm{B}}^{\mathsf{fake}}(y') = \tilde{\eta}|_{\mathrm{B}}^{\mathsf{fake}}(y')$ for all $y, y' \in \Upsilon_{\mathrm{B}}$, and for each $(y, b) \in \Upsilon_{\mathrm{B}} \times \Omega_{\mathrm{B}}$ at least one of the following two equalities holds true:

$$\tilde{\eta}((\Upsilon_{\mathrm{A}}, \Omega_{\mathrm{A}}), (y, b), (\Upsilon_{\mathrm{B}}, \Omega_{\mathrm{B}})) = \tilde{\eta}((\Upsilon_{\mathrm{A}}, \Omega_{\mathrm{A}}), (y, b), (y, b))$$
$$\tilde{\eta}((\Upsilon_{\mathrm{A}}, \Omega_{\mathrm{A}}), (\Upsilon_{\mathrm{B}}, \Omega_{\mathrm{B}}), (y, b)) = \tilde{\eta}((\Upsilon_{\mathrm{A}}, \Omega_{\mathrm{A}}), (y, b), (y, b))$$

   The intuition behind calling a cheating situation "direct" is that Bob does not sometimes claim an actual input-output tuple (y, b) to be (y', b') and also sometimes claim an actual input-output tuple (y', b') to be (y'', b''), but instead always goes the direct way: He claims (y, b) to be (y'', b'') in the first place and is just honest about (y', b').
2. Every direct cheating situation $\tilde{\eta} \in \mathfrak{N}_{\mathrm{B}}^{(F)}$ is a convex combination of a *trivial* and a *straight* cheating situation $\bar{\eta}, \hat{\eta} \in \mathfrak{N}_{\mathrm{B}}^{(F)}$, in the sense that $\bar{\eta}((\Upsilon_{\mathrm{A}}, \Omega_{\mathrm{A}}), (y, b), (y', b')) = 0$ for all distinct $(y, b), (y', b') \in \Upsilon_{\mathrm{B}} \times \Omega_{\mathrm{B}}$, and for each $y \in \Upsilon$ at least one of the following two equalities holds true:

$$\hat{\eta}|_{\mathrm{B}}^{\mathsf{true}}(y) = 0 \qquad \text{or} \qquad \hat{\eta}|_{\mathrm{B}}^{\mathsf{fake}}(y) = 0$$

   The intuition behind trivial cheating situations is that Bob is simply always honest, and the intuition behind straight cheating situations is that Bob always claims to have used some input symbol that he actually did never use at all.

This yields a more abstract redundancy criterion (Corollary 18), which plays a major role in proving existence of appropriate OT-cores for secure generation of correlated data. Moreover, this insight also helps proving that the redundancy-free version of any $F \in \mathfrak{F}_{\mathsf{fin}}$ is unique up to isomorphism (Corollary 19).

Notation 12 (Equivalent cheating situations). Let any $F \in \mathfrak{F}_{\mathsf{fin}}$ be given and $\eta, \eta' \in \mathfrak{N}_{\mathrm{B}}^{(F)}$, such that $\eta|_{\mathrm{B}}^{\mathsf{true}} = \eta'|_{\mathrm{B}}^{\mathsf{true}}$ and $\eta|_{\mathrm{B}}^{\mathsf{fake}} = \eta'|_{\mathrm{B}}^{\mathsf{fake}}$. Then η and η' are called *equivalent*, what we denote by $\eta \sim \eta'$.

Remark 13. As a direct consequence of the conditions 2 and 3 of Definition 5, every cheating situation is equivalent to its normalized version (cf. Corollary 9).

Notation 14 (Containedness). Let $F := (\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Omega_{\mathrm{A}}, \Omega_{\mathrm{B}}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$ and let $\eta, \eta' \in \mathfrak{N}_{\mathrm{B}}^{(F)}$, such that for all $x \in \Upsilon_{\mathrm{A}}$, $a \in \Omega_{\mathrm{A}}$, $y, y' \in \Upsilon_{\mathrm{B}}$, $b, b' \in \Omega_{\mathrm{B}}$ the following implication holds true:

$$\eta'((x, a), (y, b), (y', b')) > 0 \quad \Rightarrow \quad \eta((x, a), (y, b), (y', b')) > 0$$

Then we say that η *contains* η' and we denote that by $\eta \sqsupseteq \eta'$. Let $\eta \sqsupset \eta'$ denote that $\eta \sqsupseteq \eta' \not\sqsupseteq \eta$.

Lemma 15 (Generality of direct cheating situations). *Let any $F := (\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Omega_{\mathrm{A}}, \Omega_{\mathrm{B}}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$ and $\tilde{\eta} \in \mathfrak{N}_{\mathrm{B}}^{(F)}$ be given. Then there exists a direct cheating situation $\hat{\eta} \in \mathfrak{N}_{\mathrm{B}}^{(F)}$, such that $\hat{\eta} \sim \tilde{\eta}$.*

Proof. First note that by Remark 13, w.l.o.g. η is normalized. Since otherwise we just can set $\hat{\eta} := \tilde{\eta}$, w.l.o.g. we find some $\tilde{y}, \tilde{y}' \in \Upsilon_{\mathrm{B}}$, $\tilde{b}, \tilde{b}' \in \Omega_{\mathrm{B}}$, such that $(\tilde{y}, \tilde{b}) \neq (\tilde{y}', \tilde{b}')$ and:

$$\tilde{\eta}((\Upsilon_{\mathrm{A}}, \Omega_{\mathrm{A}}), (\Upsilon_{\mathrm{B}}, \Omega_{\mathrm{B}}), (\tilde{y}, \tilde{b})) > \tilde{\eta}((\Upsilon_{\mathrm{A}}, \Omega_{\mathrm{A}}), (\tilde{y}, \tilde{b}), (\tilde{y}, \tilde{b}))$$
$$\tilde{\eta}((\Upsilon_{\mathrm{A}}, \Omega_{\mathrm{A}}), (\tilde{y}, \tilde{b}), (\tilde{y}', \tilde{b}')) > 0$$

We will now construct a normalized cheating situation $\tilde{\eta}' \in \mathfrak{N}_{\mathrm{B}}^{(F)}$ with the following properties.

(**a**) It holds that $\tilde{\eta}' \sim \tilde{\eta}$.

(**b**) At least one of the following two equalities does hold true:

$$\tilde{\eta}'((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(\Upsilon_\mathrm{B},\Omega_\mathrm{B}),(\tilde{y},\tilde{b})) = \tilde{\eta}'((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(\tilde{y},\tilde{b}),(\tilde{y},\tilde{b}))$$
$$\tilde{\eta}'((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(\tilde{y},\tilde{b}),(\tilde{y}',\tilde{b}')) = 0$$

(**c**) For all $y' \in \Upsilon_\mathrm{B}$, $b' \in \Omega_\mathrm{B}$ with $(\tilde{y},\tilde{b}) \neq (y',b')$ and $\tilde{\eta}((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(\tilde{y},\tilde{b}),(y',b')) = 0$ it still does hold that $\tilde{\eta}'((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(\tilde{y},\tilde{b}),(y',b')) = 0$.

(**d**) For all $y \in \Upsilon_\mathrm{B}$, $b \in \Omega_\mathrm{B}$ with $\tilde{\eta}((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(\Upsilon_\mathrm{B},\Omega_\mathrm{B}),(y,b)) = \tilde{\eta}((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(y,b),(y,b))$ it still does hold that $\tilde{\eta}'((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(\Upsilon_\mathrm{B},\Omega_\mathrm{B}),(y,b)) = \tilde{\eta}'((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(y,b),(y,b))$.

Our lemma then follows by induction. For our construction of $\tilde{\eta}'$ we first define the auxiliary values $\gamma,\gamma',\delta \in \mathbb{R}_{>0}$ as follows:

$$\gamma := \tilde{\eta}((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(\Upsilon_\mathrm{B},\Omega_\mathrm{B}),(\tilde{y},\tilde{b})) - \tilde{\eta}((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(\tilde{y},\tilde{b}),(\tilde{y},\tilde{b}))$$
$$\gamma' := \tilde{\eta}((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(\tilde{y},\tilde{b}),(\tilde{y}',\tilde{b}'))$$
$$\delta := \min(\gamma,\gamma')$$

Now we define the mapping $\Delta : \Upsilon_\mathrm{A}\times\Omega_\mathrm{A}\times(\Upsilon_\mathrm{B}\times\Omega_\mathrm{B})^2 \to \mathbb{R}$ by:

$$\Delta((x,a),(y,b),(y',b')) := \begin{cases} -\frac{\delta\cdot\tilde{\eta}((x,a),(y,b),(\tilde{y},\tilde{b}))}{\gamma} & \text{if } (y,b)\neq(\tilde{y},\tilde{b}) \text{ and } (y',b')=(\tilde{y},\tilde{b}) \\ \frac{\delta\cdot\tilde{\eta}((x,a),(y,b),(\tilde{y},\tilde{b}))}{\gamma} & \text{if } (y,b)\neq(\tilde{y},\tilde{b}) \text{ and } (y',b')=(\tilde{y}',\tilde{b}') \\ -\frac{\delta\cdot\tilde{\eta}((x,a),(\tilde{y},\tilde{b}),(\Upsilon_\mathrm{B},\Omega_\mathrm{B}))}{\tilde{\eta}((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(\tilde{y},\tilde{b}),(\Upsilon_\mathrm{B},\Omega_\mathrm{B}))} & \text{if } (y,b)=(\tilde{y},\tilde{b}) \text{ and } (y',b')=(\tilde{y}',\tilde{b}') \\ \frac{\delta\cdot\tilde{\eta}((x,a),(\tilde{y},\tilde{b}),(\Upsilon_\mathrm{B},\Omega_\mathrm{B}))}{\tilde{\eta}((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(\tilde{y},\tilde{b}),(\Upsilon_\mathrm{B},\Omega_\mathrm{B}))} & \text{if } (y,b)=(\tilde{y},\tilde{b}) \text{ and } (y',b')=(\tilde{y},\tilde{b}) \\ 0 & \text{else} \end{cases}$$

Since $\tilde{\eta}$ is normalized, it is straightforward to verify that $\Delta((x,a),(\Upsilon_\mathrm{B},\Omega_\mathrm{B}),(y',b')) = 0$ and $\Delta((x,a),(y,b),(\Upsilon_\mathrm{B},\Omega_\mathrm{B})) = 0$ for all $x \in \Upsilon_\mathrm{A}$, $a \in \Omega_\mathrm{A}$, $y,y' \in \Upsilon_\mathrm{B}$, $b,b' \in \Omega_\mathrm{B}$. Hence, the mapping $\tilde{\eta}' := \tilde{\eta} + \Delta$ fulfills the conditions 1–3 of Definition 5, and $\tilde{\eta}' \sim \tilde{\eta}$. Further, by Condition 4 of Definition 5 one can conclude quite easily that $\Delta((x,a),(y,b),(y',b')) \geq -\tilde{\eta}((x,a),(y,b),(y',b'))$ for all $x \in \Upsilon_\mathrm{A}$, $a \in \Omega_\mathrm{A}$, $y,y' \in \Upsilon_\mathrm{B}$, $b,b' \in \Omega_\mathrm{B}$ and therefore $\tilde{\eta}'((x,a),(y,b),(y',b')) \geq 0$. Finally, by a simple case analysis one can show that for all $x \in \Upsilon_\mathrm{A}$, $a \in \Omega_\mathrm{A}$, $y,y' \in \Upsilon_\mathrm{B}$, $b,b' \in \Omega_\mathrm{B}$ with $\tilde{\eta}((x,a),(y,b),(\Upsilon_\mathrm{B},\Omega_\mathrm{B})) > 0$ it holds:

$$\frac{\Delta((x,a),(y,b),(y',b'))}{\tilde{\eta}((x,a),(y,b),(\Upsilon_\mathrm{B},\Omega_\mathrm{B}))} = \frac{\Delta((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(y,b),(y',b'))}{\tilde{\eta}((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(y,b),(\Upsilon_\mathrm{B},\Omega_\mathrm{B}))}$$

Thereby, since $\Delta((x,a),(y,b),(\Upsilon_\mathrm{B},\Omega_\mathrm{B})) = 0$, for all $x \in \Upsilon_\mathrm{A}$, $a \in \Omega_\mathrm{A}$, $y,y' \in \Upsilon_\mathrm{B}$, $b,b' \in \Omega_\mathrm{B}$ with $\tilde{\eta}'((x,a),(y,b),(\Upsilon_\mathrm{B},\Omega_\mathrm{B})) > 0$ follows:

$$\frac{\tilde{\eta}'((x,a),(y,b),(y',b'))}{\tilde{\eta}'((x,a),(y,b),(\Upsilon_\mathrm{B},\Omega_\mathrm{B}))} = \frac{\tilde{\eta}'((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(y,b),(y',b'))}{\tilde{\eta}'((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(y,b),(\Upsilon_\mathrm{B},\Omega_\mathrm{B}))}$$

Thus, $\tilde{\eta}'$ is a normalized cheating situation for F with $\tilde{\eta}' \sim \tilde{\eta}$. Yet, there are just the properties (**b**), (**c**) and (**d**) left to prove.

*Proof for (**b**):* The property (**b**) follows by our choice of δ. In the case that $\delta = \gamma$, we have that $\Delta((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(y,b),(\tilde{y},\tilde{b})) = -\tilde{\eta}((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(y,b),(\tilde{y},\tilde{b}))$ for all $(y,b) \in (\Upsilon_\mathrm{B}\times\Omega_\mathrm{B})\setminus\{(\tilde{y},\tilde{b})\}$, whereby follows that $\tilde{\eta}'((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(\Upsilon_\mathrm{B},\Omega_\mathrm{B}),(\tilde{y},\tilde{b})) = \tilde{\eta}'((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(\tilde{y},\tilde{b}),(\tilde{y},\tilde{b}))$. In the case that $\delta = \gamma'$, we have that $\Delta((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(\tilde{y},\tilde{b}),(\tilde{y}',\tilde{b}')) = -\tilde{\eta}((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(\tilde{y},\tilde{b}),(\tilde{y}',\tilde{b}'))$, whereby follows that $\tilde{\eta}'((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(\tilde{y},\tilde{b}),(\tilde{y}',\tilde{b}')) = 0$.

*Proof for (**c**):* By construction of Δ, for all $x \in \Upsilon_{\mathrm{B}}$, $a \in \Omega_{\mathrm{B}}$, $y' \in \Upsilon_{\mathrm{B}}$, $b' \in \Omega_{\mathrm{B}}$ with $(y', b') \neq (\tilde{y}, \tilde{b})$ it holds that $\Delta\big((x,a),(\tilde{y},\tilde{b}),(y',b')\big) \leq 0$, what yields:

$$\tilde{\eta}'\big((x,a),(\tilde{y},\tilde{b}),(y',b')\big) \leq \tilde{\eta}\big((x,a),(\tilde{y},\tilde{b}),(y',b')\big)$$

*Proof for (**d**):* Let us assume that we could find some $y \in \Upsilon_{\mathrm{B}}$, $b \in \Omega_{\mathrm{B}}$ with:

$$\tilde{\eta}\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big) = \tilde{\eta}\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(y,b),(y,b)\big)$$
$$\tilde{\eta}'\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big) > \tilde{\eta}'\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(y,b),(y,b)\big)$$

This would directly yield that $\Delta\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big) > \Delta\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(y,b),(y,b)\big)$, but by construction of Δ for all $x \in \Upsilon_{\mathrm{A}}$, $a \in \Omega_{\mathrm{A}}$, $y \in \Upsilon_{\mathrm{B}}$, $b \in \Omega_{\mathrm{B}}$ it holds:

$$\Delta\big((x,a),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big) = 0$$
$$\Delta\big((x,a),(y,b),(y,b)\big) \geq 0$$

□

Lemma 16 (Decomposition of direct cheating situations). *Let $F := (\Upsilon_{\mathrm{A}},\Upsilon_{\mathrm{B}},\Omega_{\mathrm{A}},\Omega_{\mathrm{B}},\phi) \in \mathfrak{F}_{\mathsf{fin}}$ and let $\eta \in \mathfrak{N}_{\mathrm{B}}^{(F)}$ be direct. Then η is straight or it contains a trivial cheating situation for F.*

Proof. W.l.o.g. we assume that η is not straight, i.e. we find some $\tilde{y} \in \Upsilon_{\mathrm{B}}$, such that $\eta|^{\mathsf{true}}(\tilde{y}) > 0$ and $\eta|^{\mathsf{fake}}(\tilde{y}) > 0$. We now construct a trivial cheating situation $\tilde{\eta}$, such that $\tilde{\eta} \sqsubseteq \eta$. We define the following mapping:

$$\tilde{\eta} : (\Upsilon_{\mathrm{A}}\times\Omega_{\mathrm{A}})\times(\Upsilon_{\mathrm{B}}\times\Omega_{\mathrm{B}})^2 \to \mathbb{R}_{\geq 0},\ \big((x,a),(y,b),(y',b')\big) \mapsto \begin{cases} \frac{\phi_{x,\tilde{y}}(a,b)}{|\Upsilon_{\mathrm{A}}|} & \text{if } y = y' = \tilde{y} \text{ and } b = b' \\ 0 & \text{else} \end{cases}$$

It is pretty obvious that $\tilde{\eta}$ is a trivial cheating situation for F. So there is just left to show that $\tilde{\eta} \sqsubseteq \eta$. So, let some arbitrary $x \in \Upsilon_{\mathrm{A}}$, $a \in \Omega_{\mathrm{A}}$, $y, y' \in \Upsilon_{\mathrm{B}}$, $b, b' \in \Omega_{\mathrm{B}}$ be given with:

$$\tilde{\eta}\big((x,a),(y,b),(y',b')\big) > 0$$

By construction of $\tilde{\eta}$ this means that $y = y' = \tilde{y}$ and $b = b'$ and $\phi_{x,\tilde{y}}(a,b) > 0$. By our choice of $\tilde{y}$ and the conditions 2 and 3 of Definition 5 follows:

$$\eta\big((x,a),(\tilde{y},b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big) > 0$$
$$\eta\big((x,a),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}}),(\tilde{y},b)\big) > 0$$

Since η is direct by assumption, this implies that $\eta\big((x,a),(\tilde{y},b),(\tilde{y},b)\big) > 0$. Since $y = y' = \tilde{y}$ and $b = b'$, this means that $\eta\big((x,a),(y,b),(y',b')\big) > 0$. This is what we had to show. □

Corollary 17 (General decomposition of cheating situations). *Let $F := (\Upsilon_{\mathrm{A}},\Upsilon_{\mathrm{B}},\Omega_{\mathrm{A}},\Omega_{\mathrm{B}},\phi) \in \mathfrak{F}_{\mathsf{fin}}$ and let $\eta \in \mathfrak{N}_{\mathrm{B}}^{(F)}$. Then η is equivalent to a trivial cheating situation for F or there exists a convex combination $\eta' := t\cdot\bar{\eta} + (1-t)\cdot\hat{\eta}$ of a trivial cheating situation $\bar{\eta} \in \mathfrak{N}_{\mathrm{B}}^{(F)}$ and a straight cheating situation $\hat{\eta} \in \mathfrak{N}_{\mathrm{B}}^{(F)}$, such that $\eta \sim \eta'$.*

Proof. W.l.o.g. we assume that η is a direct cheating situation (cf. Lemma 15). Further, w.l.o.g. we assume that η is neither trivial nor straight. We will now construct some cheating situations $\bar{\eta}', \hat{\eta}' \in \mathfrak{N}_{\mathrm{B}}^{(F)}$ meeting the following four conditions:

$\bar{\eta}' \sqsubseteq \eta$ $\qquad$ $\hat{\eta}' \sqsubset \eta$ $\qquad$ $\bar{\eta}'$ is trivial $\qquad$ η is a convex combination of $\bar{\eta}'$ and $\hat{\eta}'$

Since every convex combination of trivial cheating situations for F itself is a trivial cheating situation, our lemma then follows by induction.

By Lemma 16, we find some trivial cheating situation $\bar{\eta}' \in \mathfrak{N}_{\mathrm{B}}^{(F)}$, such that $\bar{\eta}' \sqsubseteq \eta$. Now, let $t := \max\{s \in \mathbb{R} \mid \mathrm{Image}(\eta - s\cdot\bar{\eta}') \subseteq \mathbb{R}_{\geq 0}\}$. Note that $0 < t < 1$ by our choice of η and $\bar{\eta}'$. We set $\hat{\eta}' := (1-t)^{-1}\cdot(\eta - t\cdot\bar{\eta}')$. By Lemma 10, it follows that $\hat{\eta}' \in \mathfrak{N}_{\mathrm{B}}^{(F)}$. Moreover, by our choice of t we have that $\hat{\eta}' \sqsubset \eta$. □

Corollary 18 (Characteristic-based redundancy criterion). *Let* $F := (\Upsilon_\mathrm{A}, \Upsilon_\mathrm{B}, \Omega_\mathrm{A}, \Omega_\mathrm{B}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$ *and let* $y' \in \Upsilon_\mathrm{B}$, $\xi \in \mathfrak{X}_\mathrm{B}^{(F)}$, *such that* $\xi(y) \leq 0$ *for all* $y \in \Upsilon_\mathrm{B} \backslash \{y'\}$ *and* $\xi(y') > 0$. *Then* y' *is redundant.*

Proof. By Definition 7 we find some normalized cheating situation $\eta \in \mathfrak{N}_\mathrm{B}^{(F)}$, such that for all $y \in \Upsilon_\mathrm{B}$ it holds:

$$\eta|_\mathrm{B}^{\mathsf{fake}}(y) > \eta|_\mathrm{B}^{\mathsf{true}}(y) \text{ if } y = y'$$
$$\eta|_\mathrm{B}^{\mathsf{fake}}(y) \leq \eta|_\mathrm{B}^{\mathsf{true}}(y) \text{ else}$$

By Corollary 17 we can choose η to be straight (we just discard the trivial part), whereby it follows:

$$\eta|_\mathrm{B}^{\mathsf{fake}}(y') = 1 \qquad \text{and} \qquad \eta|_\mathrm{B}^{\mathsf{true}}(y') = 0$$

Now, let $\tau := \eta|_\mathrm{B}^{\mathsf{true}}$. We also find some family of probability mass functions $\lambda := (\lambda_{y,b})_{y \in \Upsilon_\mathrm{B}, b \in \Omega_\mathrm{B}} \subseteq \mathrm{pmf}(\Omega_\mathrm{B})$, such that for all $y \in \Upsilon_\mathrm{B}$, $b, b' \in \Omega_\mathrm{B}$ with $\eta((\Upsilon_\mathrm{A}, \Omega_\mathrm{A}), (y, b), (\Upsilon_\mathrm{B}, \Omega_\mathrm{B})) > 0$ we have:

$$\lambda_{y,b}(b') = \frac{\eta((\Upsilon_\mathrm{A}, \Omega_\mathrm{A}), (y, b), (y', b'))}{\eta((\Upsilon_\mathrm{A}, \Omega_\mathrm{A}), (y, b), (\Upsilon_\mathrm{B}, \Omega_\mathrm{B}))}$$

Exploiting the conditions 2, 4 and 3 of Definition 5, we get for all $x \in \Upsilon_\mathrm{A}$, $a \in \Omega_\mathrm{A}$, $b' \in \Omega_\mathrm{B}$:

$$\sum\nolimits_{y \in \Upsilon_\mathrm{B},\, b \in \Omega_\mathrm{B}} \tau(y) \cdot \phi_{x,y}(a, b) \cdot \lambda_{y,b}(b') \;=\; \sum\nolimits_{y \in \Upsilon_\mathrm{B},\, b \in \Omega_\mathrm{B}} \frac{\eta((x, a), (y, b), (\Upsilon_\mathrm{B}, \Omega_\mathrm{B})) \cdot \lambda_{y,b}(b')}{\eta|_\mathrm{A}(x)}$$
$$= \sum\nolimits_{y \in \Upsilon_\mathrm{B},\, b \in \Omega_\mathrm{B}} \frac{\eta((x, a), (y, b), (y', b'))}{\eta|_\mathrm{A}(x)} \;=\; \eta|_\mathrm{B}^{\mathsf{fake}}(y') \cdot \phi_{x,y'}(a, b') \;=\; \phi_{x,y'}(a, b') \qquad \square$$

Corollary 19 (Uniqueness of redundancy-free versions). *Let any* $F := (\Upsilon_\mathrm{A}, \Upsilon_\mathrm{B}, \Omega_\mathrm{A}, \Omega_\mathrm{B}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$ *be given. Then the redundancy-free version of* F *is unique up to isomorphism.*

Proof. We have to show: For any two redundant input symbols $\tilde{y}', \tilde{y}'' \in \Upsilon_\mathrm{B}$, such that after removing one of them the other is not redundant any more, the respective block columns in the condensed canonical representation are equal up to internal permutation of columns. This implies that it does not matter in which order redundant input symbols are removed from Υ_B (or Υ_A respectively).

So, let any such $\tilde{y}', \tilde{y}'' \in \Upsilon_\mathrm{B}$ be given. Since $\tilde{y}'$ is redundant, we find some $\tau \in \mathrm{pmf}(\Upsilon_\mathrm{B})$ and $\{\lambda_{y,b}\}_{y \in \Upsilon_\mathrm{B}, b \in \Omega_\mathrm{B}} \subseteq \mathrm{pmf}(\Omega_\mathrm{B})$, such that $\tau(\tilde{y}') = 0$ and for all $x \in \Upsilon_\mathrm{A}$, $a \in \Omega_\mathrm{A}$, $b' \in \Omega_\mathrm{B}$ it holds:

$$\phi_{x,\tilde{y}'}(a, b') = \sum\nolimits_{y \in \Upsilon_\mathrm{B},\, b \in \Omega_\mathrm{B}} \tau(y) \cdot \phi_{x,y}(a, b) \cdot \lambda_{y,b}(b')$$

Thus, we can construct a (normalized) cheating situation as follows:

$$\eta'((x, a), (y, b), (y', b')) := \begin{cases} \frac{1}{|\Upsilon_\mathrm{A}|} \cdot \tau(y) \cdot \phi_{x,y}(a, b) \cdot \lambda_{y,b}(b') & \text{if } y \neq \tilde{y}' \text{ and } y' = \tilde{y}' \\ 0 & \text{else} \end{cases}$$

Note that by construction η' is straight and $\eta'|_\mathrm{B}^{\mathsf{fake}}(\tilde{y}') = 1$. Further note that $\eta'|_\mathrm{B}^{\mathsf{true}}(\tilde{y}'') > 0$ by our choice of $\tilde{y}', \tilde{y}''$. Analogously, we find some straight cheating situation $\eta'' \in \mathfrak{B}_\mathrm{B}^{(F)}$, such that $\eta''|_\mathrm{B}^{\mathsf{fake}}(\tilde{y}'') = 1$ and $\eta''|_\mathrm{B}^{\mathsf{true}}(\tilde{y}') > 0$. Let $t := \eta''|_\mathrm{B}^{\mathsf{true}}(\tilde{y}')$. Now, by Lemma 10, we can construct a new cheating strategy η as follows:

$$\eta := \tfrac{1}{1+t} \cdot \eta'' + \tfrac{t}{1+t} \cdot \eta'$$

By construction we have that $\eta|_\mathrm{B}^{\mathsf{fake}}(\{\tilde{y}', \tilde{y}''\}) = 1$ and $\eta|_\mathrm{B}^{\mathsf{fake}}(\tilde{y}') = \eta|_\mathrm{B}^{\mathsf{true}}(\tilde{y}') = \frac{t}{1+t}$. By Corollary 17 we can conclude that η is either equivalent to a trivial cheating situation, or there exists a straight cheating situation $\hat{\eta} \in \mathfrak{B}_\mathrm{B}^{(F)}$ such that $\hat{\eta}|_\mathrm{B}^{\mathsf{fake}}(\tilde{y}'') = 1$ and $\hat{\eta}|_\mathrm{B}^{\mathsf{fake}}(\tilde{y}') = \hat{\eta}|_\mathrm{B}^{\mathsf{true}}(\tilde{y}') = 0$. Since the latter is ruled out by our choice of $\tilde{y}', \tilde{y}''$, we have that $\eta|_\mathrm{B}^{\mathsf{fake}}(\tilde{y}'') = \eta|_\mathrm{B}^{\mathsf{true}}(\tilde{y}'') = \frac{1}{1+t}$, whereby it follows:

$$\eta'|_\mathrm{B}^{\mathsf{fake}}(\tilde{y}') = \eta'|_\mathrm{B}^{\mathsf{true}}(\tilde{y}'') = 1 \qquad \text{and} \qquad \eta''|_\mathrm{B}^{\mathsf{fake}}(\tilde{y}'') = \eta''|_\mathrm{B}^{\mathsf{true}}(\tilde{y}') = 1$$

Intuitively speaking, a corrupted Bob can replace the input symbols $\tilde{y}'$ and $\tilde{y}''$ just by each other. It is straightforward now to verify that in the condensed canonical representation of F the block columns belonging to $\tilde{y}'$ and $\tilde{y}''$ are equal up to internal permutation of columns. $\square$

4.4 Existence of robust OT-cores

In this section we show for every redundancy-free 2-party function $F \in \mathfrak{F}_{\mathsf{fin}}$ that it has some OT-core useful for us, if it has any OT-core at all. This is the core argumentation of the algebraic part of our security proof.

Definition 20 (OT-cores)**.** Let $F := (\Upsilon_\mathrm{A}, \Upsilon_\mathrm{B}, \Omega_\mathrm{A}, \Omega_\mathrm{B}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$ and some $(x,a),(x',a') \in \Upsilon_\mathrm{A} \times \Omega_\mathrm{A}$, $(y,b),(y',b') \in \Upsilon_\mathrm{B} \times \Omega_\mathrm{B}$ be given. We call $\{(x,a),(x',a')\} \times \{(y,b),(y',b')\}$ an *OT-core* of F, if $\phi_{x,y}(a,b) \cdot \phi_{x',y'}(a',b') \neq \phi_{x',y}(a',b) \cdot \phi_{x,y}(a,b')$ and at most one of the factors is zero.

Notation 21 (Hideable inputs)*.* Let $F := (\Upsilon_\mathrm{A}, \Upsilon_\mathrm{B}, \Omega_\mathrm{A}, \Omega_\mathrm{B}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$. For $Y \subseteq \Upsilon_\mathrm{B}$ we define:

$$\Psi_F(Y) \;:=\; \{y \in \Upsilon_\mathrm{B} \mid \exists\, \eta \in \mathfrak{N}_\mathrm{B}^{(F)} :\; \eta|_\mathrm{B}^{\mathsf{true}}(y) > 0 \;\wedge\; \eta|_\mathrm{B}^{\mathsf{fake}}(Y) = 1\}$$

Given any $y, y' \in \Upsilon_\mathrm{B}$, we write $\Psi_F(y,y')$ instead of $\Psi_F(\{y,y'\})$ for convenience.

Remark 22*.* Let $F := (\Upsilon_\mathrm{A}, \Upsilon_\mathrm{B}, \Omega_\mathrm{A}, \Omega_\mathrm{B}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$. Then for all $Y \subseteq \Upsilon_\mathrm{B}$ it holds:

$$\Psi_F(Y) = Y \cup \{y \in \Upsilon_\mathrm{B} \mid \exists\, \xi \in \mathfrak{X}_\mathrm{B}^{(F)} :\; \xi(y) < 0 \;\wedge\; \forall\, y' \in \Upsilon_\mathrm{B} \backslash Y : \xi(y') \leq 0\}$$

Lemma 23 (Monotonicity of Ψ_F)**.** *Let $F := (\Upsilon_\mathrm{A}, \Upsilon_\mathrm{B}, \Omega_\mathrm{A}, \Omega_\mathrm{B}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$ and let $Y' \subseteq \Upsilon_\mathrm{B}$. Then for all $Y \subseteq \Psi_F(Y')$ we have that $\Psi_F(Y) \subseteq \Psi_F(Y')$.*

Proof. Let $Y \subseteq \Psi_F(Y')$. By Remark 22 and Corollary 11 we find some cheating characteristics $\xi', \xi'' \in \mathfrak{X}_\mathrm{B}^{(F)}$, such that for all $y \in \Upsilon_\mathrm{B}$ it holds:

$$\begin{aligned} &\xi'(y) < 0 \text{ if } y \in \Psi_F(Y) \backslash Y && \qquad \xi''(y) < 0 \text{ if } y \in \Psi_F(Y') \backslash Y' \\ &\xi'(y) \leq 0 \text{ if } y \in \Upsilon_\mathrm{B} \backslash \Psi_F(Y) && \qquad \xi''(y) \leq 0 \text{ if } y \in \Upsilon_\mathrm{B} \backslash \Psi_F(Y') \end{aligned}$$

Now we find some $\gamma \in \mathbb{R}_{>0}$, such that $\gamma \cdot \xi''(y) < -\xi'(y)$ for all $y \in \Psi_F(Y') \backslash Y'$. Since $Y \subseteq \Psi_F(Y')$ by assumption, we especially have that $\gamma \cdot \xi''(y) + \xi'(y) < 0$ for all $y \in Y \backslash Y'$. Let $\xi := \xi' + \gamma \cdot \xi''$ (cf. Corollary 11). Thereby, for all $y \in \Upsilon_\mathrm{B}$ we can conclude:

$$\begin{aligned} &\xi(y) < 0 \text{ if } y \in \Psi_F(Y) \backslash Y' \\ &\xi(y) \leq 0 \text{ if } y \notin Y' \end{aligned}$$

Hence, by Remark 22 it must hold that $\Psi_F(Y) \subseteq \Psi_F(Y')$. □

Lemma 24. *Let $F := (\Upsilon_\mathrm{A}, \Upsilon_\mathrm{B}, \Omega_\mathrm{A}, \Omega_\mathrm{B}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$, such that Υ_B does not contain any redundant input symbols. Further let $\tilde{y}, \tilde{y}' \in \Upsilon_\mathrm{B}$ and $\tilde{Y}, \tilde{Y}' \subseteq \Psi_F(\tilde{y}, \tilde{y}')$, such that $\Psi_F(\tilde{y}, \tilde{y}') \supsetneq \{\tilde{y}, \tilde{y}'\}$ and for all $\hat{y} \in \tilde{Y}$, $\hat{y}' \in \tilde{Y}'$ it holds that $\Psi_F(\hat{y}, \hat{y}') = \Psi_F(\tilde{y}, \tilde{y}')$. Then for all $\hat{y} \in \tilde{Y}$, $\hat{y}' \in \tilde{Y}'$, $Y \subseteq \tilde{Y} \cup \tilde{Y}'$ with $\hat{y}, \hat{y}' \notin Y$ there exists some $\xi \in \mathfrak{X}_\mathrm{B}^{(F)}$, such that for all $y \in \Upsilon_\mathrm{B}$ it holds:*

$$\begin{aligned} &\xi(y) > 0 \text{ if } y \in \{\hat{y}, \hat{y}'\} \\ &\xi(y) = 0 \text{ if } y \in Y \text{ or } y \notin \Psi_F(\tilde{y}, \tilde{y}') \\ &\xi(y) < 0 \text{ else} \end{aligned}$$

Proof. Our proof is by induction on $|Y|$. So in the first instance we assume that $Y = \emptyset$. Let $\hat{y} \in \tilde{Y}$ and $\hat{y}' \in \tilde{Y}'$ be arbitrary. Since $\Psi_F(\hat{y}, \hat{y}') = \Psi_F(\tilde{y}, \tilde{y}')$ by assumption, we find some $\xi \in \mathfrak{X}_\mathrm{B}^{(F)}$ by Remark 22 and Corollary 11, such that for all $y \in \Upsilon_\mathrm{B}$ it holds:

$$\begin{aligned} &\xi(y) = 0 \text{ if } y \notin \Psi_F(\tilde{y}, \tilde{y}') \\ &\xi(y) < 0 \text{ if } y \in \Psi_F(\tilde{y}, \tilde{y}') \backslash \{\hat{y}, \hat{y}'\} \end{aligned}$$

Moreover, since $\Psi_F(\tilde{y},\tilde{y}')\setminus\{\hat{y},\hat{y}'\} \neq \emptyset$ by assumption and $\xi(\Upsilon_\mathrm{B}) = 0$ by definition, it must hold that $\xi(\hat{y}) > 0$; else $\hat{y}'$ would be redundant by Corollary 18. Analogously it follows that $\xi(\hat{y}') > 0$. Thereby we have proven our lemma for the case that $|Y| = 0$.

Now, let $Y \subseteq \tilde{Y} \cup \tilde{Y}'$ with $Y \neq \emptyset$ and let $\hat{y} \in \tilde{Y}$, $\hat{y}' \in \tilde{Y}'$, such that $\hat{y}, \hat{y}' \notin Y$. Furthermore, let $y' \in Y$. There are two cases to be considered: $y' \in \tilde{Y}$ and $y' \in \tilde{Y}'$. Since both cases can be handled analogously, we just consider the latter. By induction hypothesis we find some $\xi', \xi'' \in \mathfrak{X}_\mathrm{B}^{(F)}$, such that for all $y \in \Upsilon_\mathrm{B}$ it holds:

$$\begin{array}{ll}
\xi'(y) > 0 \text{ if } y \in \{\hat{y}, y'\} & \xi''(y) > 0 \text{ if } y \in \{\hat{y}, \hat{y}'\} \\
\xi'(y) = 0 \text{ if } y \in Y\setminus\{y'\} \text{ or } y \notin \Psi_F(\tilde{y},\tilde{y}') & \xi''(y) = 0 \text{ if } y \in Y\setminus\{y'\} \text{ or } y \notin \Psi_F(\tilde{y},\tilde{y}') \\
\xi'(y) < 0 \text{ else} & \xi''(y) < 0 \text{ else}
\end{array}$$

We set $\xi := \xi'(y')\cdot\xi'' - \xi''(y')\cdot\xi'$; note that $\xi \in \mathfrak{X}_\mathrm{B}^{(F)}$ by Corollary 11, since $\xi'(y') > 0$ and $\xi''(y') < 0$. By construction, for all $y \in \Upsilon_\mathrm{B}$ it follows:

$$\begin{array}{l}
\xi(y) > 0 \text{ if } y = \hat{y} \\
\xi(y) = 0 \text{ if } y \in Y \text{ or } y \notin \Psi_F(\tilde{y},\tilde{y}') \\
\xi(y) < 0 \text{ if } y \in \Psi_F(\tilde{y},\tilde{y}') \text{ and } y \notin Y \cup \{\hat{y},\hat{y}'\}
\end{array}$$

Finally, we still must have that $\xi(\hat{y}') > 0$, since otherwise $\hat{y}$ would be redundant by Corollary 18. □

Lemma 25. *Let $F := (\Upsilon_\mathrm{A}, \Upsilon_\mathrm{B}, \Omega_\mathrm{A}, \Omega_\mathrm{B}, \phi) \in \mathfrak{F}_\mathsf{fin}$, such that Υ_B does not contain any redundant input symbols. Further let $\{(\tilde{x},\tilde{a}),(\tilde{x}',\tilde{a}')\}\times\{(\tilde{y},\tilde{b}),(\tilde{y}',\tilde{b}')\} \subseteq (\Upsilon_\mathrm{A}\times\Omega_\mathrm{A})^2\times(\Upsilon_\mathrm{B}\times\Omega_\mathrm{B})^2$ be an OT-core of F. Then there exist some $(\bar{y},\bar{b}),(\bar{y}',\bar{b}') \in \Upsilon_\mathrm{B}\times\Omega_\mathrm{B}$, such that $\{(\tilde{x},\tilde{a}),(\tilde{x}',\tilde{a}')\}\times\{(\bar{y},\bar{b}),(\bar{y}',\bar{b}')\}$ also is an OT-core of F and $\Psi_F(\bar{y},\bar{y}') = \{\bar{y},\bar{y}'\}$.*

Proof. W.l.o.g., $\phi_{\tilde{x},\tilde{y}}(\tilde{a},\tilde{b}) > 0$ and $\phi_{\tilde{x}',\tilde{y}}(\tilde{a}',\tilde{b}) > 0$, i.e. we can write $\frac{\phi_{\tilde{x},\tilde{y}'}(\tilde{a},\tilde{b}')}{\phi_{\tilde{x},\tilde{y}}(\tilde{a},\tilde{b})} \neq \frac{\phi_{\tilde{x}',\tilde{y}'}(\tilde{a}',\tilde{b}')}{\phi_{\tilde{x}',\tilde{y}}(\tilde{a}',\tilde{b})}$; else we interchange $(\tilde{y},\tilde{b})$ and $(\tilde{y}',\tilde{b}')$. We define the following input sets:

$$\begin{aligned}
\hat{Y}' &:= \{y' \in \Upsilon_\mathrm{B} \mid \text{for some } b' \in \Omega_\mathrm{B},\ \{(\tilde{x},\tilde{a}),(\tilde{x}',\tilde{a}')\}\times\{(\tilde{y},\tilde{b}),(y',b')\} \text{ is an OT-core of } F\} \\
\hat{Y} &:= \{y \in \Upsilon_\mathrm{B}\setminus\hat{Y}' \mid \text{for some } b \in \Omega_\mathrm{B},\ \{(\tilde{x},\tilde{a}),(\tilde{x}',\tilde{a}')\}\times\{(y,b),(\tilde{y}',\tilde{b}')\} \text{ is an OT-core of } F\} \\
\hat{Y}_0 &:= \{y_0 \in \Upsilon_\mathrm{B} \mid \phi_{\tilde{x},y_0}(\tilde{a},\Omega_\mathrm{B}) = \phi_{\tilde{x}',y_0}(\tilde{a}',\Omega_\mathrm{B}) = 0\}
\end{aligned}$$

Further we set $\tilde{Y} := \hat{Y} \cap \Psi_F(\tilde{y},\tilde{y}')$ and $\tilde{Y}' := \hat{Y}' \cap \Psi_F(\tilde{y},\tilde{y}')$ and $\tilde{Y}_0 := \hat{Y}_0 \cap \Psi_F(\tilde{y},\tilde{y}')$. Note that $\hat{Y} \cup \hat{Y}' \cup \hat{Y}_0 = \Upsilon_\mathrm{B}$ and thereby $\tilde{Y} \cup \tilde{Y}' \cup \tilde{Y}_0 = \Psi_F(\tilde{y},\tilde{y}')$. Further note that $\tilde{y} \in \tilde{Y}$ and $\tilde{y}' \in \tilde{Y}'$.

We prove our lemma by contradiction. So, we assume that our initially given OT-core is a minimal counterexample in the sense that $\Psi_F(\tilde{y},\tilde{y}') \supsetneq \{\tilde{y},\tilde{y}'\}$ and for all $y \in \tilde{Y}$, $y' \in \tilde{Y}'$ it holds that $\Psi_F(y,y') = \Psi_F(\tilde{y},\tilde{y}')$ (cf. Lemma 23). Now, by Lemma 24 instantiated with $Y := (\tilde{Y} \cup \tilde{Y}')\setminus\{\tilde{y},\tilde{y}'\}$, we find some $\xi \in \mathfrak{X}_\mathrm{B}^{(F)}$, such that for all $y \in \Upsilon_\mathrm{B}$ it holds:

$$\begin{array}{l}
\xi(y) > 0 \text{ if } y \in \{\tilde{y},\tilde{y}'\} \\
\xi(y) = 0 \text{ if } y \in Y \text{ or } y \notin \Psi_F(\tilde{y},\tilde{y}') \\
\xi(y) < 0 \text{ else}
\end{array}$$

In other words, since $\Psi_F(\tilde{y},\tilde{y}') = \tilde{Y} \cup \tilde{Y}' \cup \tilde{Y}_0$ and $\tilde{Y}_0 \cap \tilde{Y} = \tilde{Y}_0 \cap \tilde{Y}' = \emptyset$, for all $y \in \Upsilon_\mathrm{B}$ it holds:

$$\begin{array}{l}
\xi(y) > 0 \text{ if } y \in \{\tilde{y},\tilde{y}'\} \\
\xi(y) < 0 \text{ if } y \in \tilde{Y}_0 \\
\xi(y) = 0 \text{ else}
\end{array}$$

Thereby we find a cheating situation $\eta \in \mathfrak{N}_{\mathrm{B}}^{(F)}$, such that for all $y \in \Upsilon_{\mathrm{B}}$ it holds:

$$\begin{aligned}\eta|_{\mathrm{B}}^{\mathsf{fake}}(y) &> \eta|_{\mathrm{B}}^{\mathsf{true}}(y) \text{ if } y \in \{\tilde{y}, \tilde{y}'\}\\ \eta|_{\mathrm{B}}^{\mathsf{fake}}(y) &< \eta|_{\mathrm{B}}^{\mathsf{true}}(y) \text{ if } y \in \tilde{Y}_0\\ \eta|_{\mathrm{B}}^{\mathsf{fake}}(y) &= \eta|_{\mathrm{B}}^{\mathsf{true}}(y) \text{ else}\end{aligned}$$

By Corollary 17 we can decompose η into a trivial and a straight part. Let $\hat{\eta}$ denote the straight part. By construction it holds:

$$\begin{aligned}\hat{\eta}|_{\mathrm{B}}^{\mathsf{true}}(\tilde{Y}_0) &= 1\\ \hat{\eta}|_{\mathrm{B}}^{\mathsf{fake}}(\tilde{y}) &> 0\end{aligned}$$

However, by our choice of $\hat{Y}_0$ we have:

$$0 \;=\; \hat{\eta}|_{\mathrm{A}}(\tilde{x}) \cdot \sum\nolimits_{y \in \hat{Y}_0} \hat{\eta}|_{\mathrm{B}}^{\mathsf{true}}(y) \cdot \phi_{\tilde{x},y}(\tilde{a}, \Omega_{\mathrm{B}}) \;\geq\; \hat{\eta}|_{\mathrm{A}}(\tilde{x}) \cdot \sum\nolimits_{y \in \tilde{Y}_0} \hat{\eta}|_{\mathrm{B}}^{\mathsf{true}}(y) \cdot \phi_{\tilde{x},y}(\tilde{a}, \Omega_{\mathrm{B}})$$

Hence, by Condition 2 of Definition 5 we can conclude that $0 \geq \eta((\tilde{x}, \tilde{a}), (\tilde{Y}_0, \Omega_{\mathrm{B}}), (\Upsilon_{\mathrm{B}}, \Omega_{\mathrm{B}}))$. Because $\hat{\eta}|_{\mathrm{B}}^{\mathsf{true}}(\tilde{Y}_0) = 1$ and thus $\hat{\eta}((\tilde{x}, \tilde{a}), (\Upsilon_{\mathrm{B}} \backslash \tilde{Y}_0, \Omega_{\mathrm{B}}), (\Upsilon_{\mathrm{B}}, \Omega_{\mathrm{B}})) = 0$, we also have:

$$\hat{\eta}((\tilde{x}, \tilde{a}), (\tilde{Y}_0, \Omega_{\mathrm{B}}), (\Upsilon_{\mathrm{B}}, \Omega_{\mathrm{B}})) \;=\; \hat{\eta}((\tilde{x}, \tilde{a}), (\Upsilon_{\mathrm{B}}, \Omega_{\mathrm{B}}), (\Upsilon_{\mathrm{B}}, \Omega_{\mathrm{B}})) \;\geq\; \hat{\eta}((\tilde{x}, \tilde{a}), (\Upsilon_{\mathrm{B}}, \Omega_{\mathrm{B}}), (\tilde{y}, \tilde{b}))$$

Now, since $\phi_{\tilde{x},\tilde{y}}(\tilde{a}, \tilde{b}) > 0$ by assumption and we found $\hat{\eta}|_{\mathrm{B}}^{\mathsf{fake}}(\tilde{y}) > 0$, we can finally estimate by the conditions 1 and 2 of Definition 5:

$$\hat{\eta}((\tilde{x}, \tilde{a}), (\Upsilon_{\mathrm{B}}, \Omega_{\mathrm{B}}), (\tilde{y}, \tilde{b})) = \hat{\eta}|_{\mathrm{A}}(\tilde{x}) \cdot \hat{\eta}|_{\mathrm{B}}^{\mathsf{fake}}(\tilde{y}) \cdot \phi_{\tilde{x},\tilde{y}}(\tilde{a}, \tilde{b}) > 0$$

Putting things together, we get the contradiction that $0 > 0$, what concludes our proof. □

Corollary 26 (Existence of robust OT-cores). *Let $F := (\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Omega_{\mathrm{A}}, \Omega_{\mathrm{B}}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$, such that Υ_{B} does not contain any redundant input symbols. Further let $\{(\tilde{x}, \tilde{a}), (\tilde{x}', \tilde{a}')\} \times \{(\tilde{y}, \tilde{b}), (\tilde{y}', \tilde{b}')\} \subseteq (\Upsilon_{\mathrm{A}} \times \Omega_{\mathrm{A}})^2 \times (\Upsilon_{\mathrm{B}} \times \Omega_{\mathrm{B}})^2$ be an OT-core of F. Then there exist some $(\bar{y}, \bar{b}), (\bar{y}', \bar{b}') \in \Upsilon_{\mathrm{B}} \times \Omega_{\mathrm{B}}$, such that $\{(\tilde{x}, \tilde{a}), (\tilde{x}', \tilde{a}')\} \times \{(\bar{y}, \bar{b}), (\bar{y}', \bar{b}')\}$ also is an OT-core of F and for every cheating situation $\eta \in \mathfrak{N}_{\mathrm{B}}^{(F)}$ with $\eta|_{\mathrm{B}}^{\mathsf{fake}}(\{\bar{y}, \bar{y}'\}) = 1$ and all $y \in \Upsilon_{\mathrm{B}}$ it holds:*

$$\eta|_{\mathrm{B}}^{\mathsf{fake}}(y) \;=\; \eta|_{\mathrm{B}}^{\mathsf{true}}(y)$$

Proof. By Lemma 25 we find some $(\bar{y}, \bar{b}), (\bar{y}', \bar{b}') \in \Upsilon_{\mathrm{B}} \times \Omega_{\mathrm{B}}$, such that $\{(\tilde{x}, \tilde{a}), (\tilde{x}', \tilde{a}')\} \times \{(\bar{y}, \bar{b}), (\bar{y}', \bar{b}')\}$ also is an OT-core of F and $\Psi_F(\bar{y}, \bar{y}') = \{\bar{y}, \bar{y}'\}$. Now, let any $\eta \in \mathfrak{N}_{\mathrm{B}}^{(F)}$ with $\eta|_{\mathrm{B}}^{\mathsf{fake}}(\{\bar{y}, \bar{y}'\}) = 1$ be given. We just have to show:

$$\begin{aligned}\eta|_{\mathrm{B}}^{\mathsf{fake}}(\bar{y}) &= \eta|_{\mathrm{B}}^{\mathsf{true}}(\bar{y})\\ \eta|_{\mathrm{B}}^{\mathsf{fake}}(\bar{y}') &= \eta|_{\mathrm{B}}^{\mathsf{true}}(\bar{y}')\end{aligned}$$

We pick the following cheating characteristic (q.v. Definition 7):

$$\xi : \Upsilon_{\mathrm{B}} \to \mathbb{R},\; y \mapsto \eta|_{\mathrm{B}}^{\mathsf{fake}}(y) - \eta|_{\mathrm{B}}^{\mathsf{true}}(y)$$

Since $\eta|_{\mathrm{B}}^{\mathsf{fake}}(\{\bar{y}, \bar{y}'\}) = 1$ and thus $\eta|_{\mathrm{B}}^{\mathsf{fake}}(\Upsilon_{\mathrm{B}} \backslash \{\bar{y}, \bar{y}'\}) = 0$, for all $y \in \Upsilon_{\mathrm{B}} \backslash \{\bar{y}, \bar{y}'\}$ it must hold that $\xi(y) \leq 0$. Moreover, since $\Psi_F(\bar{y}, \bar{y}') = \{\bar{y}, \bar{y}'\}$, for all $y \in \Upsilon_{\mathrm{B}} \backslash \{\bar{y}, \bar{y}'\}$ we actually have that $\xi(y) = 0$ by Remark 22. Since $\xi(\Upsilon_{\mathrm{B}}) = 0$, it follows that $\xi(\bar{y}) = -\xi(\bar{y}')$. Now, if $\xi(\bar{y}) \neq 0$, this would render either $\bar{y}$ or $\bar{y}'$ redundant by Corollary 18. Thus, it must hold that $\xi(\bar{y}) = \xi(\bar{y}') = 0$. □

Protocol $\pi_F(X, Y, \alpha, \beta, \gamma)$

Parametrized by a 2-party function $F := (\Upsilon_\mathrm{A}, \Upsilon_\mathrm{B}, \Omega_\mathrm{A}, \Omega_\mathrm{B}, \phi) \in \mathfrak{F}_\mathsf{fin}$ and $(X, Y, \alpha, \beta, \gamma) \in \Pi_F$. Let k denote the security parameter and let $K := \{1, \ldots, k\}$. The protocol proceeds as follows.

0. **Initialization:** Alice initializes two empty strings $s_\mathrm{A}^\mathsf{in}, s_\mathrm{A}^\mathsf{out}$ and an index set $K_\mathrm{A} \leftarrow K$. Bob analogously initializes $s_\mathrm{B}^\mathsf{in}, s_\mathrm{B}^\mathsf{out}, K_\mathrm{B}$. Let the probability mass functions $\tilde{n}_\mathrm{A}, \tilde{n}_\mathrm{B}$ be defined by:

$$\tilde{n}_\mathrm{A} : \Upsilon_\mathrm{A} \to \mathbb{R}_{>0},\ x \mapsto \begin{cases} (1-k^{-\alpha}) \cdot |X|^{-1} + k^{-\alpha} \cdot |\Upsilon_\mathrm{A}|^{-1} & \text{if } x \in X \\ k^{-\alpha} \cdot |\Upsilon_\mathrm{A}|^{-1} & \text{else} \end{cases}$$

$$\tilde{n}_\mathrm{B} : \Upsilon_\mathrm{B} \to \mathbb{R}_{>0},\ y \mapsto \begin{cases} (1-k^{-\alpha}) \cdot |Y|^{-1} + k^{-\alpha} \cdot |\Upsilon_\mathrm{B}|^{-1} & \text{if } y \in Y \\ k^{-\alpha} \cdot |\Upsilon_\mathrm{B}|^{-1} & \text{else} \end{cases}$$

1. **Invocation of F:** According to $\tilde{n}_\mathrm{A}$ Alice randomly chooses some input symbol $x \in \Upsilon_\mathrm{A}$; Bob randomly chooses some $y \in \Upsilon_\mathrm{B}$ according to $\tilde{n}_\mathrm{B}$. Then F is invoked with the input tuple (x, y), i.e. Alice learns some $a \in \Omega_\mathrm{A}$ and Bob learns some $b \in \Omega_\mathrm{B}$ with (a, b) distributed according to $\phi_{x,y}$. Alice concatenates x to s_A^in and a to $s_\mathrm{A}^\mathsf{out}$ respectively; Bob concatenates y to s_B^in and b to $s_\mathrm{B}^\mathsf{out}$.

   This protocol step is executed for k times.

2. **Check A:** Alice picks some uniformly random index set $\bar{K}_\mathrm{A} \subseteq K_\mathrm{A}$ with[a] $|\bar{K}_\mathrm{A}| = k^{\frac{1}{2}+\beta}$ and sends $\bar{K}_\mathrm{A}$ to Bob, who announces $\big(\hat{s}_\mathrm{B}^\mathsf{in}[\bar{K}_\mathrm{A}], \hat{s}_\mathrm{B}^\mathsf{out}[\bar{K}_\mathrm{A}]\big) := \big(s_\mathrm{B}^\mathsf{in}[\bar{K}_\mathrm{A}], s_\mathrm{B}^\mathsf{out}[\bar{K}_\mathrm{A}]\big)$. Alice aborts the protocol, if she finds some $x \in \Upsilon_\mathrm{A}$, $y \in \Upsilon_\mathrm{B}$, $a \in \Omega_\mathrm{A}$, $b \in \Omega_\mathrm{B}$ with:

$$\big|s_\mathrm{A}^\mathsf{in} \times s_\mathrm{A}^\mathsf{out} \times \hat{s}_\mathrm{B}^\mathsf{in} \times \hat{s}_\mathrm{B}^\mathsf{out}[\bar{K}_\mathrm{A}]\big|_{(x,a,y,b)} \neq k^{\frac{1}{2}+\beta} \cdot \tilde{n}_\mathrm{A}(x) \cdot \tilde{n}_\mathrm{B}(y) \cdot \phi_{x,y}(a,b) \pm k^{\frac{1}{4}+\beta}$$

   At the end of this protocol step, Alice sets $K_\mathrm{A} \leftarrow K_\mathrm{A} \backslash \bar{K}_\mathrm{A}$ and Bob sets $K_\mathrm{B} \leftarrow K_\mathrm{B} \backslash \bar{K}_\mathrm{A}$.

3. **Check B:** This protocol step proceeds analogously to Check A with interchanged roles of Alice and Bob.

4. **Output:** Alice announces the index set $K'_\mathrm{A} := \big\{i \in K_\mathrm{A} \;\big|\; s_\mathrm{A}^\mathsf{in}[i] \in X\big\}$, then Bob announces $K'_\mathrm{B} := \big\{i \in K_\mathrm{B} \;\big|\; s_\mathrm{B}^\mathsf{in}[i] \in Y\big\}$; let $K' := K'_\mathrm{A} \cap K'_\mathrm{B}$. If $|K'| < k - k^{1-\gamma}$, the protocol is aborted; else Alice outputs the compound string $s_\mathrm{A}^\mathsf{in} \times s_\mathrm{A}^\mathsf{out}[K']$ and Bob outputs $s_\mathrm{B}^\mathsf{in} \times s_\mathrm{B}^\mathsf{out}[K']$.

[a] W.l.o.g. we have that $k^{\frac{1}{2}+\beta} \in \mathbb{N}$, since $\beta \in \mathbb{Q}$ and w.l.o.g. $k \in \big\{l^\zeta \;\big|\; l, \zeta \in \mathbb{N}, \text{ such that } \zeta \cdot (\frac{1}{2}+\beta) \in \mathbb{N}\big\}$.

Figure 4.1: Protocol scheme for secure generation of correlated data from a given 2-party function.

4.5 Protocol for generation of correlated data

Now we give the formal description of our generic protocol scheme for generation of correlated data (q.v. Figure 4.1). For convenience, we use the following quite self-suggesting notations.

Notation 27. Let "$a = b \pm c$" denote that $|a - b| < |c|$, i.e. the value a differs from b at most by c.

Notation 28. Let s be a finite string over some alphabet Ω. By $|s|$ we denote the length of s. By $|s|_\alpha$ with $\alpha \in \Omega$ we denote the number of appearances of α in s. We canonically extend this notation to subalphabets $T \subseteq \Omega$ by $|s|_T := \sum_{\alpha \in T} |s|_\alpha$. By $s[i]$ with $i \in \{1, \ldots, |s|\}$ we denote the i-th element of s. For $n \in \mathbb{N}$ and a given index set $K = \{k_1, \ldots, k_n\} \subset \mathbb{N}$ with $0 < k_1 < \ldots < k_n \leq |s|$, we denote the string $s[k_1]\, s[k_2] \ldots s[k_n]$ by $s[k_1, \ldots, k_n]$, or simply by $s[K]$. Further, for some given strings s_A and s_B of the same length, we define the *compound string* $s_\mathrm{A} \times s_\mathrm{B}$, whose i-th element is just the tuple $\big(s_\mathrm{A}[i], s_\mathrm{B}[i]\big)$. We denote the i-th element of such a compound string by $s_\mathrm{A} \times s_\mathrm{B}[i]$.

Notation 29. Given any $F := (\Upsilon_\mathrm{A}, \Upsilon_\mathrm{B}, \Omega_\mathrm{A}, \Omega_\mathrm{B}, \phi) \in \mathfrak{F}_\mathsf{fin}$, let Π_F denote the set of all quintuples $(X, Y, \alpha, \beta, \gamma)$, where $X \subseteq \Upsilon_\mathrm{A}$, $Y \subseteq \Upsilon_\mathrm{B}$, $\alpha, \beta, \gamma \in \mathbb{R}_{>0}$, such that $X, Y \neq \emptyset$ and $\beta \in \mathbb{Q}$ with $\beta < \frac{1}{6}$.

Notation 30. Given any $F \in \mathfrak{F}_\mathsf{fin}$ and $(X, Y, \alpha, \beta, \gamma) \in \Pi_F$, we define the following characteristics

for non-aborted protocol runs of $\pi_F(X,Y,\alpha,\beta,\gamma)$:

$$\nu_{\mathrm{B}}\big((x,a),(y,b),(y',b')\big) := \frac{\Big|s_{\mathrm{A}}^{\mathsf{in}} \times s_{\mathrm{A}}^{\mathsf{out}} \times s_{\mathrm{B}}^{\mathsf{in}} \times s_{\mathrm{B}}^{\mathsf{out}} \times \hat{s}_{\mathrm{B}}^{\mathsf{in}} \times \hat{s}_{\mathrm{B}}^{\mathsf{out}}[\bar{K}_{\mathrm{A}}]\Big|_{(x,a,y,b,y',b')}}{\Big|\bar{K}_{\mathrm{A}}\Big|}$$

$$\nu_{\mathrm{A}}\big((x,a),(x',a'),(y,b)\big) := \frac{\Big|s_{\mathrm{A}}^{\mathsf{in}} \times s_{\mathrm{A}}^{\mathsf{out}} \times \hat{s}_{\mathrm{A}}^{\mathsf{in}} \times \hat{s}_{\mathrm{A}}^{\mathsf{out}} \times s_{\mathrm{B}}^{\mathsf{in}} \times s_{\mathrm{B}}^{\mathsf{out}}[\bar{K}_{\mathrm{B}}]\Big|_{(x,a,x',a'y,b)}}{\Big|\bar{K}_{\mathrm{B}}\Big|}$$

For convenience, we also write:

$$\nu_{\mathrm{B}}|_{\mathrm{A}}(x) := \nu_{\mathrm{B}}\big((x,\Omega_{\mathrm{A}}),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}}),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big) \qquad \nu_{\mathrm{A}}|_{\mathrm{B}}(y) := \nu_{\mathrm{A}}\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}}),(y,\Omega_{\mathrm{B}})\big)$$
$$\nu_{\mathrm{B}}|_{\mathrm{B}}^{\mathsf{true}}(y) := \nu_{\mathrm{B}}\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(y,\Omega_{\mathrm{B}}),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big) \qquad \nu_{\mathrm{A}}|_{\mathrm{A}}^{\mathsf{true}}(x) := \nu_{\mathrm{A}}\big((x,\Omega_{\mathrm{A}}),(\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big)$$
$$\nu_{\mathrm{B}}|_{\mathrm{B}}^{\mathsf{fake}}(y') := \nu_{\mathrm{B}}\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}}),(y',\Omega_{\mathrm{B}})\big) \qquad \nu_{\mathrm{A}}|_{\mathrm{A}}^{\mathsf{fake}}(x') := \nu_{\mathrm{A}}\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(x',\Omega_{\mathrm{A}}),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big)$$

4.6 Real protocol runs versus idealized cheating situations

We show now that our notion of cheating situations is close to what may ever happen during real protocol runs. Therefor, we utilize some powerful tools from probability theory (Lemma 31 and Corollary 32) and real algebraic geometry (Lemma 35 and Corollary 36). The former are borrowed from [KMQ10], but the latter are completely novel tools, which were necessary because our notion of cheating situations is more complex than that of [KMQ10, KMQ11]. In particular, normalized cheating situations in [KMQ10] can be described by linear constraints, which is not true in our case due to Condition 4 of Definition 5.

Lemma 31 (Stability of random distributions, [KMQ10, Lemma 15]). *Let some sequence $(\mathbf{x}_k)_{k\in\mathbb{N}}$ of binomially and/or hypergeometrically distributed random variables $\mathbf{x}_k$ be given, such that $\mathbb{P}[0\leq\mathbf{x}_k\leq k]=1$ for all $k\in\mathbb{N}$. Further let $\Delta>\frac{1}{2}$. Then the probability $\mathbb{P}\big[|\mathbf{x}_k-\mathbb{E}(\mathbf{x}_k)|\geq k^{\Delta}\big]$ is negligible in k.*

Proof. By [Hoe63, Theorem 2], for all $n\in\mathbb{N}$, $c\in\mathbb{R}_{>0}$ and every binomially distributed random variable $\mathbf{x}$ with $\mathbb{P}[0\leq\mathbf{x}\leq n]=1$ it holds that $\mathbb{P}\big[|\mathbf{x}-\mathbb{E}(\mathbf{x})|\geq c\big]\leq 2\cdot\exp\left(-2c^2\cdot n^{-1}\right)$. In [Hoe63, Section 6] it was shown that this estimation holds for hypergeometrically distributed X, too. For all $k\in\mathbb{N}$, we can conclude:

$$\mathbb{P}\big[|\mathbf{x}_k-\mathbb{E}(\mathbf{x}_k)|\geq k^{\Delta}\big] \;\leq\; 2\cdot\exp\left(-2k^{2\Delta-1}\right) \qquad \square$$

Corollary 32 ([KMQ10, Corollary 16]). *Let $\mathcal{H}$ be some memoryless random source that samples from some finite alphabet Ω. Let $p:\Omega\to\mathbb{R}$, $x\mapsto\mathbb{P}[\mathcal{H}\text{ outputs }x]$. Further let $\mathcal{A}$ be some arbitrary algorithm that on input $k\in\mathbb{N}$ sequentially samples up to k random symbols $\mathbf{x}_1,\ldots,\mathbf{x}_{\mathbf{n}}\leftarrow\mathcal{H}$, i.e. $\mathbf{n}$ is a random variable with $\mathbb{P}[1\leq\mathbf{n}\leq k]=1$ and $\mathbf{n}$ may be correlated with $(\mathbf{x}_1,\ldots,\mathbf{x}_{\mathbf{n}})$. Then for all constants $\Delta>\frac{1}{2}$ and all $S\subseteq\Omega$ the probability $\mathbb{P}\Big[\big||\mathbf{x}_1\ldots\mathbf{x}_{\mathbf{n}}|_S-\mathbf{n}\cdot p(S)\big|\geq k^{\Delta}\Big]$ is negligible in k.*

Proof. For our proof we make $\mathcal{A}$ a bit more powerful: $\mathcal{A}$ always samples exactly k random symbols $\mathbf{x}_1,\ldots,\mathbf{x}_k\leftarrow\mathcal{H}$ and then computes and outputs $\mathbf{n}$.

Now, for $n\in\{1,\ldots,k\}$, $S\subseteq\Omega$ let $\mathbf{X}_n(S):=|\mathbf{x}_1\ldots\mathbf{x}_n|_S$. Analogously to the proof of Lemma 31, for all $n\in\{0,\ldots,k\}$ and $S\subseteq\Omega$ it always holds:

$$\mathbb{P}\big[|\mathbf{X}_n(S)-n\cdot p(S)|\geq k^{\Delta}\big] \;\leq\; \mathbb{P}\big[|\mathbf{X}_n(S)-n\cdot p(S)|\geq n^{\Delta}\big] \;\leq\; 2\cdot\exp(-2n^{2\Delta-1})$$

Further, for $n<k^{\Delta}$ it trivially holds that $\mathbb{P}\big[|\mathbf{X}_n(S)-n\cdot p(S)|\geq k^{\Delta}\big]=0$. Hence follows:

$$\mathbb{P}\big[|\mathbf{X}_{\mathbf{n}}(S)-\mathbf{n}\cdot p(S)|\geq k^{\Delta}\big] \;\leq\; \sum_{n=\lceil k^{\Delta}\rceil}^{k}\mathbb{P}\big[|\mathbf{X}_n(S)-n\cdot p(S)|\geq k^{\Delta}\big] \;\leq\; \frac{2(k-k^{\Delta})}{\exp\big(2k^{\Delta(2\Delta-1)}\big)} \qquad \square$$

Corollary 33. *Let some arbitrary* $F := (\Upsilon_\mathrm{A}, \Upsilon_\mathrm{B}, \Omega_\mathrm{A}, \Omega_\mathrm{B}, \phi) \in \mathfrak{F}_\mathsf{fin}$ *and* $(X, Y, \alpha, \beta, \gamma) \in \Pi_F$ *be given, as well as some constant* $\Delta > \frac{1}{2}$*. Let* $\pi := \pi_F(X, Y, \alpha, \beta, \gamma)$ *and let* k *denote the security parameter. Then, if Alice is honest, a protocol run of* π *with overwhelming probability is either aborted or for all* $x \in \Upsilon_\mathrm{A}$, $a \in \Omega_\mathrm{A}$, $y \in \Upsilon_\mathrm{B}$ *and* $b \in \Omega_\mathrm{B}$ *we have that* $\nu_\mathrm{B}\big((x,a),(y,b),(\Upsilon_\mathrm{B},\Omega_\mathrm{B})\big) = k^{-1}\big|s_\mathrm{A}^\mathsf{in} \times s_\mathrm{A}^\mathsf{out} \times s_\mathrm{B}^\mathsf{in} \times s_\mathrm{B}^\mathsf{out}\big|_{(x,a,y,b)} \pm k^{\Delta-(\frac{1}{2}+\beta)}$*. If Bob is honest, the analog holds for* ν_A*.*

Proof. Let us consider some arbitrary but fixed $x \in \Upsilon_\mathrm{A}$, $a \in \Omega_\mathrm{A}$, $y \in \Upsilon_\mathrm{B}$ and $b \in \Omega_\mathrm{B}$. Once the compound string $s_\mathrm{A}^\mathsf{in} \times s_\mathrm{A}^\mathsf{out} \times s_\mathrm{B}^\mathsf{in} \times s_\mathrm{B}^\mathsf{out}$ is generated by Alice and Bob calling F in the protocol step Invocation of F, we can consider an honest Alice's random choice of $\bar{K}_\mathrm{A}$ as a random experiment with hypergeometrically distributed outcome $\big|s_\mathrm{A}^\mathsf{in} \times s_\mathrm{A}^\mathsf{out} \times s_\mathrm{B}^\mathsf{in} \times s_\mathrm{B}^\mathsf{out}[\bar{K}_\mathrm{A}]\big|_{(x,a,y,b)}$. Now, by Lemma 31 we have with overwhelming probability:

$$\big|s_\mathrm{A}^\mathsf{in} \times s_\mathrm{A}^\mathsf{out} \times s_\mathrm{B}^\mathsf{in} \times s_\mathrm{B}^\mathsf{out}[\bar{K}_\mathrm{A}]\big|_{(x,a,y,b)} \;=\; k^{\frac{1}{2}+\beta} \cdot \frac{\big|s_\mathrm{A}^\mathsf{in} \times s_\mathrm{A}^\mathsf{out} \times s_\mathrm{B}^\mathsf{in} \times s_\mathrm{B}^\mathsf{out}\big|_{(x,a,y,b)}}{k} \pm k^{\Delta}$$

As $\nu_\mathrm{B}\big((x,a),(y,b),(\Upsilon_\mathrm{B},\Omega_\mathrm{B})\big) = k^{-(\frac{1}{2}+\beta)}\big|s_\mathrm{A}^\mathsf{in} \times s_\mathrm{A}^\mathsf{out} \times s_\mathrm{B}^\mathsf{in} \times s_\mathrm{B}^\mathsf{out}[\bar{K}_\mathrm{A}]\big|_{(x,a,y,b)}$ by definition (cf. Notation 30), we can conclude:

$$\nu_\mathrm{B}\big((x,a),(y,b),(\Upsilon_\mathrm{B},\Omega_\mathrm{B})\big) \;=\; k^{-1}\big|s_\mathrm{A}^\mathsf{in} \times s_\mathrm{A}^\mathsf{out} \times s_\mathrm{B}^\mathsf{in} \times s_\mathrm{B}^\mathsf{out}\big|_{(x,a,y,b)} \pm k^{\Delta-(\frac{1}{2}+\beta)}$$

If Bob is honest, we have to take into account that a corrupted Alice might choose $\bar{K}_\mathrm{A}$ maliciously and thereby introduce an additional error of at most $k^{\beta-\frac{1}{2}}$ into our estimation, i.e.:

$$\nu_\mathrm{A}\big((x,a),(\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(y,b)\big) \;=\; k^{-1}\big|s_\mathrm{A}^\mathsf{in} \times s_\mathrm{A}^\mathsf{out} \times s_\mathrm{B}^\mathsf{in} \times s_\mathrm{B}^\mathsf{out}\big|_{(x,a,y,b)} \pm \big(k^{\Delta-(\frac{1}{2}+\beta)} + k^{\beta-\frac{1}{2}}\big)$$

However, since $\beta < \frac{1}{6}$ by definition (cf. Notation 29) and the estimation also holds for any Δ' with $\frac{1}{2} < \Delta' < \Delta$, we can argue:

$$k^{\Delta'-(\frac{1}{2}+\beta)} + k^{\beta-\frac{1}{2}} \;=\; k^{\Delta-(\frac{1}{2}+\beta)} \cdot k^{\Delta'-\Delta} \cdot \big(1 + k^{-\Delta'+2\beta}\big) \;<\; k^{\Delta-(\frac{1}{2}+\beta)} \cdot \underbrace{k^{\Delta'-\Delta} \cdot \big(1 + k^{-\frac{1}{6}}\big)}_{\leq 1 \text{ for almost all } k}$$

Hence, if Bob is honest, a protocol run of π with overwhelming probability is either aborted or for all $x \in \Upsilon_\mathrm{A}$, $a \in \Omega_\mathrm{A}$, $y \in \Upsilon_\mathrm{B}$ and $b \in \Omega_\mathrm{B}$ we have:

$$\nu_\mathrm{A}\big((x,a),(\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(y,b)\big) = k^{-1}\big|s_\mathrm{A}^\mathsf{in} \times s_\mathrm{A}^\mathsf{out} \times s_\mathrm{B}^\mathsf{in} \times s_\mathrm{B}^\mathsf{out}\big|_{(x,a,y,b)} \pm k^{\Delta-(\frac{1}{2}+\beta)} \qquad \square$$

Lemma 34. *Let some arbitrary* $F := (\Upsilon_\mathrm{A}, \Upsilon_\mathrm{B}, \Omega_\mathrm{A}, \Omega_\mathrm{B}, \phi) \in \mathfrak{F}_\mathsf{fin}$ *and* $(X, Y, \alpha, \beta, \gamma) \in \Pi_F$ *be given, as well as some constant* $\Delta > \frac{1}{2}$*. Let* $\pi := \pi_F(X, Y, \alpha, \beta, \gamma)$ *and let* k *denote the security parameter. Then, if Alice is honest, a protocol run of* π *with overwhelming probability is either aborted or we have:*

1. *For all* $x \in \Upsilon_\mathrm{A}$ *it holds that* $\nu_\mathrm{B}|_\mathrm{A}(x) = \tilde{n}_A(x) \pm k^{(\frac{1}{2}+\beta)(\Delta-1)}$*.*
2. *For all* $x \in \Upsilon_\mathrm{A}$, $a \in \Omega_\mathrm{A}$, $y \in \Upsilon_\mathrm{B}$, $b \in \Omega_\mathrm{B}$ *it holds:*

$$\nu_\mathrm{B}\big((x,a),(y,b),(\Upsilon_\mathrm{B},\Omega_\mathrm{B})\big) = \nu_\mathrm{B}|_\mathrm{A}(x) \cdot \nu_\mathrm{B}|_\mathrm{B}^\mathsf{true}(y) \cdot \phi_{x,y}(a,b) \pm k^{(\frac{1}{2}+\beta)(\Delta-1)}$$

3. *For all* $x \in \Upsilon_\mathrm{A}$, $a \in \Omega_\mathrm{A}$, $y' \in \Upsilon_\mathrm{B}$, $b' \in \Omega_\mathrm{B}$ *it holds:*

$$\nu_\mathrm{B}\big((x,a),(\Upsilon_\mathrm{B},\Omega_\mathrm{B}),(y',b')\big) = \nu_\mathrm{B}|_\mathrm{A}(x) \cdot \nu_\mathrm{B}|_\mathrm{B}^\mathsf{fake}(y') \cdot \phi_{x,y}(a,b) \pm k^{\frac{\Delta-1}{2}}$$

4. *For all* $x \in \Upsilon_\mathrm{A}$, $a \in \Omega_\mathrm{A}$, $y, y' \in \Upsilon_\mathrm{B}$, $b, b' \in \Omega_\mathrm{B}$ *with* $\nu_\mathrm{B}\big((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(y,b),(\Upsilon_\mathrm{B},\Omega_\mathrm{B})\big) > 0$ *it holds:*

$$\nu_\mathrm{B}\big((x,a),(y,b),(y',b')\big) = \nu_\mathrm{B}\big((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(y,b),(y',b')\big) \cdot \tfrac{\nu_\mathrm{B}((x,a),(y,b),(\Upsilon_\mathrm{B},\Omega_\mathrm{B}))}{\nu_\mathrm{B}((\Upsilon_\mathrm{A},\Omega_\mathrm{A}),(y,b),(\Upsilon_\mathrm{B},\Omega_\mathrm{B}))} \pm k^{(\frac{1}{2}+\beta)(\Delta-1)}$$

If Bob is honest, the analog holds for ν_{A}.

Proof. The techniques needed here are pretty much the same as for the proof of Corollary 33. The assertions 1 and 4 are direct consequences of Lemma 31, as we can consider $k^{\frac{1}{2}+\beta} \cdot \nu_{\mathrm{B}}|_{\mathrm{A}}(x)$ a binomially distributed random variable and $k^{\frac{1}{2}+\beta} \cdot \nu_{\mathrm{B}}\big((x,a),(y,b),(y',b')\big)$ a hypergeometrically distributed random variable in the respective context. Assertion 2 can be derived from Corollary 32, as a corrupted Bob's view in the protocol step Invocation of F can be seen as maliciously sampling from $|\Upsilon_{\mathrm{B}}|$ mutually independent memoryless random sources. Finally, an honest Alice enforces assertion 3 in the protocol step Check A, what can be shown as follows. Alice directly enforces that $\nu_{\mathrm{B}}\big((x,a),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}}),(y',b')\big) = \tilde{n}_{\mathrm{A}}(x) \cdot \tilde{n}_{\mathrm{B}}(y') \cdot \phi_{x,y'}(a,b') \pm k^{-\frac{1}{4}}$ for all $x \in \Upsilon_{\mathrm{A}}$, $a \in \Omega_{\mathrm{A}}$, $y' \in \Upsilon_{\mathrm{B}}$, $b' \in \Omega_{\mathrm{B}}$, whereby especially follows:

$$\nu_{\mathrm{B}}|_{\mathrm{A}}(x) = \tilde{n}_{\mathrm{A}}(x) \pm |\Omega_{\mathrm{A}} \times \Omega_{\mathrm{B}}| \cdot k^{-\frac{1}{4}}$$
$$\nu_{\mathrm{B}}|_{\mathrm{B}}^{\mathsf{fake}}(y') = \tilde{n}_{\mathrm{B}}(y') \pm |\Omega_{\mathrm{A}} \times \Omega_{\mathrm{B}}| \cdot k^{-\frac{1}{4}}$$

Thereby, we already have:

$$\nu_{\mathrm{B}}\big((x,a),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}}),(y',b')\big) = \nu_{\mathrm{B}}|_{\mathrm{A}}(x) \cdot \nu_{\mathrm{B}}|_{\mathrm{B}}^{\mathsf{fake}}(y') \cdot \phi_{x,y'}(a,b') \pm \big(2 \cdot |\Omega_{\mathrm{A}} \times \Omega_{\mathrm{B}}| + 1\big)k^{-\frac{1}{4}}$$

Furthermore, we can estimate the error term $\big(2 \cdot |\Omega_{\mathrm{A}} \times \Omega_{\mathrm{B}}| + 1\big)k^{-\frac{1}{4}}$ from above by $k^{\frac{\Delta-1}{2}}$ for almost all $k \in \mathbb{N}$, since $\Delta > \frac{1}{2}$ by assumption.

If Bob is honest, we have to take into account that a corrupted Alice might choose $\bar{K}_{\mathrm{A}}$ maliciously. This will only introduce an additional error of at most $k^{\beta-\frac{1}{2}}$ in the analog of our estimations for the assertions 1 and 2, i.e. there we formally have to replace the error term $k^{(\frac{1}{2}+\beta)(\Delta-1)}$ by $\big(k^{(\frac{1}{2}+\beta)(\Delta-1)} + k^{\beta-\frac{1}{2}}\big)$. However, since $\beta < \frac{1}{6}$ by definition (cf. Notation 29) and our estimations also hold for any Δ' with $\frac{1}{2} < \Delta' < \Delta$, we can argue:

$$\begin{aligned} k^{(\frac{1}{2}+\beta)(\Delta'-1)} + k^{\beta-\frac{1}{2}} &= k^{(\frac{1}{2}+\beta)(\Delta-1)} \cdot \big(k^{(\frac{1}{2}+\beta)(\Delta'-\Delta)} + k^{-\frac{\Delta}{2}+(2-\Delta)\beta}\big) \\ &< k^{(\frac{1}{2}+\beta)(\Delta-1)} \cdot \underbrace{\big(k^{\frac{1}{3}(\Delta'-\Delta)} + k^{-\frac{\Delta}{2}+\frac{1}{4}}\big)}_{\leq 1 \text{ for almost all } k} \end{aligned}$$

□

Lemma 35. *Let $n \in \mathbb{N}$ and some polynomial $f \in \mathbb{R}[X_1,\ldots,X_n]$ be given, such that the variety $V := \{x \in \mathbb{R}^n \mid f(x) = 0\}$ is not empty. Furthermore, let a bounded convex polytope $P \subset \mathbb{R}^n$ be given, such that $V \cap P \neq \emptyset$. Then for every norm there exist some constants $c, \delta \in \mathbb{R}_{>0}$, such that for all $x \in P$ it holds:*

$$\min_{y \in V \cap P} \|x - y\| \;\leq\; c \cdot \big|f(x)\big|^{\delta}$$

Proof. Our proof is based on the Łojasiewicz Inequality [Łoj59, Theorem 17], by which for every open set $U \subseteq \mathbb{R}^n$, every real analytic function $h : U \to \mathbb{R}$ with non-empty zero locus Z and every compact set $K \subset U$ there exist some constants $c, \delta \in \mathbb{R}_{>0}$, such that for all $x \in K$ it holds:

$$\inf_{z \in Z} \|x - z\| \;\leq\; c \cdot \big|h(x)\big|^{\delta}$$

Note that we do not need to specify the norm used, since all norms on $\mathbb{R}^n$ are equivalent. In the following, for any $x \in \mathbb{R}^n$ and $S \subseteq \mathbb{R}^n$ let $\operatorname{dist}(x,S) := \inf_{y \in S} \|x - y\|$.

We want to prove our lemma by contradiction, i.e. we assume that for all $c, \delta \in \mathbb{R}_{>0}$ there exists some $x \in P$, such that $\operatorname{dist}(x, V \cap P) > c \cdot |f(x)|^{\delta}$. In particular, we find some sequence $(x_i)_{i \in \mathbb{N}} \subseteq P$, such that $\operatorname{dist}(x_i, V \cap P) > i \cdot \sqrt[i]{|f(x)|}$ for all $i \in \mathbb{N}$. Since P is closed and bounded and thus compact, the sequence $(x_i)_{i \in \mathbb{N}}$ has some limit point $\hat{x} \in P$. Moreover, we can choose $\hat{x}$ such that $f(\hat{x}) = 0$ and thus $\hat{x} \in V$, since otherwise we had the following contradiction:

$$\infty \;=\; \liminf_{i \in \mathbb{N}} i \cdot \sqrt[i]{|f(x_i)|} \;\leq\; \liminf_{i \in \mathbb{N}} \operatorname{dist}(x_i, V \cap P) \;\leq\; \operatorname{dist}(\hat{x}, V \cap P)$$

Hence, we have that $|f(x_i)| \leq 1$ for infinitely many $i \in \mathbb{N}$ and can discard all other members of the sequence $(x_i)_{i\in\mathbb{N}}$, while preserving the property that $\operatorname{dist}(x_i, V \cap P) > i \cdot \sqrt[i]{|f(x_i)|}$ for all $i \in \mathbb{N}$. Moreover, we can now discard *any* members of the sequence $(x_i)_{i\in\mathbb{N}}$ and still preserve that property. We will exploit this extensively. In the first instance, w.l.o.g. the whole sequence $(x_i)_{i\in\mathbb{N}}$ does converge to $\hat{x}$. Further, we find the following sequences:

- $(z_i)_{i\in\mathbb{N}} \subseteq V$, such that $\operatorname{dist}(x_i, z_i) = \operatorname{dist}(x_i, V)$
- $(w_i)_{i\in\mathbb{N}} \subseteq P$, such that w_i is a convex combination of x_i and z_i, and $\operatorname{dist}(w_i, z_i)$ is minimized

We also find some finite set of degree-one polynomials $T \subset \mathbb{R}[X_1, \ldots, X_n]$, such that we can write:

$$P \;=\; \{x \in \mathbb{R}^n \mid \max_{g\in T} g(x) \leq 0\}$$

Note that $\lim_{i\to\infty} w_i = \hat{x}$ and hence for almost all (w.l.o.g. all) $i \in \mathbb{N}$ it holds:

$$\forall\, g \in T: \qquad g(w_i) = 0 \quad \Rightarrow \quad g(\hat{x}) = 0$$

Moreover, by the Łojasiewicz Inequality there exist some constants $c', \delta' \in \mathbb{R}_{>0}$, such that for all $x \in P$ it holds:

$$\operatorname{dist}(x, V) \;\leq\; c' \cdot \big|f(x)\big|^{\delta'}$$

Hence, it must hold that $z_i \notin P$ and thus $\max_{g\in T} g(w_i) = 0$ for almost all (w.l.o.g. all) $i \in \mathbb{N}$, since otherwise we had a contradiction to our choice of $(x_i)_{i\in\mathbb{N}}$. Now, as $\max_{g\in T} g(w_i) = 0$ for all $i \in \mathbb{N}$, by a pigeonhole argument there must exist some $g \in T$, such that $g(w_i) = 0$ for infinitely many (w.l.o.g. all) $i \in \mathbb{N}$. Let $\hat{g}$ be such a polynomial. We define the affine subspace $A := \{x \in \mathbb{R}^n \mid \hat{g}(x) = 0\}$ and the polytope $Q := P \cap A$. Note that $V \cap Q \neq \emptyset$, as $\hat{x} \in V \cap Q$. Now we can utilize induction on the dimension n; or to be more precise, w.l.o.g. we may assume that n is minimal in the sense that for smaller n there would not exist any counterexample for our lemma. In particular, since for $n = 0$ our lemma is trivially true, we must have that $n > 0$. By the Triangle Inequality we can estimate:

$$\forall\, i \in \mathbb{N}: \quad \operatorname{dist}(x_i, V \cap P) \;\leq\; \operatorname{dist}(x_i, w_i) + \operatorname{dist}(w_i, V \cap Q)$$

However, since $(w_i)_{i\in\mathbb{N}} \subseteq Q \subset A$ by construction, we have that estimating $\operatorname{dist}(w_i, V \cap Q)$ is the original problem with dimension $n - 1$. Since by assumption there cannot be a counterexample for our lemma with dimension $n - 1$, we find $c'', \delta'' \in \mathbb{R}_{>0}$, such that for all $w \in Q$ it holds:

$$\operatorname{dist}(w, V \cap Q) \;\leq\; c'' \cdot \big|f(w)\big|^{\delta''}$$

Let $b := \max_{a\in P} \|\nabla f(a)\|$. For all $i \in \mathbb{N}$ it holds:

$$\begin{aligned}
\operatorname{dist}(x_i, V \cap P) \;&\leq\; \operatorname{dist}(x_i, w_i) + \operatorname{dist}(w_i, V \cap Q) \\
&\leq\; \operatorname{dist}(x_i, w_i) + c'' \cdot \big|f(w_i)\big|^{\delta''} \\
&\leq\; \operatorname{dist}(x_i, w_i) + c'' \cdot \big(|f(x_i)| + \operatorname{dist}(x_i, w_i) \cdot \max_{a\in P} \|\nabla f(a)\|\big)^{\delta''} \\
&=\; \operatorname{dist}(x_i, w_i) + c'' \cdot \big(|f(x_i)| + b \cdot \operatorname{dist}(x_i, w_i)\big)^{\delta''}
\end{aligned}$$

Now we can put things together. Since each w_i is a convex combination of the respective x_i and z_i, we can estimate:

$$\operatorname{dist}(x_i, w_i) \;\leq\; \operatorname{dist}(x_i, z_i) \;=\; \operatorname{dist}(x_i, V) \;\leq\; c' \cdot \big|f(x_i)\big|^{\delta'}$$

Thus, for all $i \in \mathbb{N}$ we have:

$$\operatorname{dist}(x_i, V \cap P) \;\leq\; c' \cdot \big|f(x_i)\big|^{\delta'} + c'' \cdot \Big(\big|f(x_i)\big| + b \cdot c' \cdot \big|f(x_i)\big|^{\delta'}\Big)^{\delta''}$$

We set $\delta := \min\{1, \delta', \delta''\}$ and $c := (c' + c'') \cdot (1 + b \cdot c')^{\delta}$. Since $\lim_{i\to\infty} f(x_i) = f(\hat{x}) = 0$, we have for almost all (w.l.o.g. all) $i \in \mathbb{N}$ that $|f(x_i)|$ is sufficiently small, so that we can estimate:

$$c' \cdot \big|f(x_i)\big|^{\delta'} + c'' \cdot \Big(\big|f(x_i)\big| + b \cdot c' \cdot \big|f(x_i)\big|^{\delta'}\Big)^{\delta''} \;\leq\; c' \cdot \big|f(x_i)\big|^{\delta} + c'' \cdot \Big(\big|f(x_i)\big| + b \cdot c' \cdot \big|f(x_i)\big|^{\delta}\Big)^{\delta}$$

In conclusion, we have for all $i \in \mathbb{N}$:

$$\begin{aligned}
\operatorname{dist}(x_i, V \cap P) \;&\leq\; c' \cdot \big|f(x_i)\big|^{\delta} + c'' \cdot \Big(\big|f(x_i)\big| + b \cdot c' \cdot \big|f(x_i)\big|^{\delta}\Big)^{\delta} \\
&\leq\; (c' + c'') \cdot \Big(\big|f(x_i)\big| + b \cdot c' \cdot \big|f(x_i)\big|^{\delta}\Big)^{\delta} \\
&\leq\; (c' + c'') \cdot \Big((1 + b \cdot c') \cdot \big|f(x_i)\big|^{\delta}\Big)^{\delta} \\
&=\; c \cdot \big|f(x_i)\big|^{\delta^2}
\end{aligned}$$

This contradicts our choice of the sequence $(x_i)_{i\in\mathbb{N}}$ and thus concludes this proof. □

Corollary 36. *Let $n \in \mathbb{N}$ and some finite set of polynomials $S \subset \mathbb{R}[X_1, \ldots, X_n]$ be given, such that the variety $V := \{x \in \mathbb{R}^n \mid \forall\, f \in S :\, f(x) = 0\}$ is not empty. Furthermore, let a bounded convex polytope $P \subset \mathbb{R}^n$ be given, such that $V \cap P \neq \emptyset$. Then for every norm there exist some constants $c, \delta \in \mathbb{R}_{>0}$, such that for all $x \in P$ it holds:*

$$\min_{y\in V\cap P} \|x - y\| \;\leq\; c \cdot \max_{f\in S} \big|f(x)\big|^{\delta}$$

Proof. We define the polynomial $g := \sum_{f\in S} f^2$, whereby we get that $V = \{x \in \mathbb{R}^n \mid g(x) = 0\}$. Now, by Lemma 35 we find some constants $c', \delta' \in \mathbb{R}_{>0}$, such that for all $x \in P$ it holds:

$$\min_{y\in V\cap P} \|x - y\| \;\leq\; c' \cdot \big|g(x)\big|^{\delta'}$$

Thus, our proof is concluded by the observation that for all $x \in \mathbb{R}^n$ we have:

$$c' \cdot \big|g(x)\big|^{\delta'} \;\leq\; c' \cdot |S| \cdot \max_{f\in S} \big|f(x)\big|^{2\delta'}$$ □

Lemma 37. *Let any $F := (\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Omega_{\mathrm{A}}, \Omega_{\mathrm{B}}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$ be given. Then, if Alice is honest, there exist some constants $\varepsilon, \varepsilon' \in \mathbb{R}_{>0}$, such that for any $\pi := \pi_F(X, Y, \alpha, \beta, \gamma)$ with $(X, Y, \alpha, \beta, \gamma) \in \Pi_F$ and $\alpha < \varepsilon'$ a protocol run of π with overwhelming probability is either aborted or we have:*

$$\exists\, \eta \in \mathfrak{N}_{\mathrm{B}}^{(F)} :\; \sum\nolimits_{x\in\Upsilon_{\mathrm{A}},\, a\in\Omega_{\mathrm{A}},\, y,y'\in\Upsilon_{\mathrm{B}},\, b,b'\in\Omega_{\mathrm{B}}} \big|\eta\big((x,a),(y,b),(y',b')\big) - \nu_{\mathrm{B}}\big((x,a),(y,b),(y',b')\big)\big| \;\leq\; \tfrac{1}{k^{\varepsilon}}$$

If Bob is honest, the analog holds for ν_{A}.

Proof. For symmetry reasons it suffices to consider the case of an honest Alice.

First note that $P := \operatorname{pmf}\big((\Upsilon_{\mathrm{A}} \times \Omega_{\mathrm{A}}) \times (\Upsilon_{\mathrm{B}} \times \Omega_{\mathrm{B}})^2\big)$ is a bounded convex polytope in the linear space $\mathbb{R}^{(\Upsilon_{\mathrm{A}}\times\Omega_{\mathrm{A}})\times(\Upsilon_{\mathrm{B}}\times\Omega_{\mathrm{B}})^2}$. Moreover, consider the variety $V \subseteq \mathbb{R}^{(\Upsilon_{\mathrm{A}}\times\Omega_{\mathrm{A}})\times(\Upsilon_{\mathrm{B}}\times\Omega_{\mathrm{B}})^2}$ defined by the following polynomial equations:

$$\begin{aligned}
&\eta\big((x,a),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big) = \eta\big((x,\Omega_{\mathrm{A}}),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}}),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big) \cdot \eta\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(y,\Omega_{\mathrm{B}}),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big) \cdot \phi_{x,y}(a,b) \\
&\eta\big((x,a),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}}),(y',b')\big) = \eta\big((x,\Omega_{\mathrm{A}}),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}}),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big) \cdot \eta\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}}),(y',\Omega_{\mathrm{B}})\big) \cdot \phi_{x,y'}(a,b') \\
&\eta\big((x,a),(y,b),(y',b')\big) \cdot \eta\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big) = \eta\big((\Upsilon_{\mathrm{A}},\Omega_{\mathrm{A}}),(y,b),(y',b')\big) \cdot \eta\big((x,a),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big)
\end{aligned}$$

Note that Bob's cheating situations for F (q.v. Definition 5) are just the mappings $\eta \in V \cap P$ with $\min_{x\in\Omega_{\mathrm{A}}} \eta\big((x,\Omega_{\mathrm{A}}),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}}),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big) > 0$, i.e. we have:

$$\mathfrak{N}_{\mathrm{B}}^{(F)} \;=\; \big\{\eta \in V \cap P \mid \forall\, x \in \Omega_{\mathrm{A}} : \eta\big((x,\Omega_{\mathrm{A}}),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}}),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big) > 0\big\} \tag{4.1}$$

Now, by Corollary 36 instantiated with the L^1-norm we find some constants $c, \delta \in \mathbb{R}_{>0}$, such that for *every* probability mass function $\tilde{\eta} \in P$ that fulfills our polynomial equations stated above up to some error ρ it holds:

$$\min_{\eta \in V \cap P} \|\eta - \tilde{\eta}\|_1 \;\leq\; c \cdot \rho^{\delta}$$

Hence by Lemma 34, with some arbitrary but constant $\Delta > \frac{1}{2}$, a protocol run of π with overwhelming probability is either aborted or there exists a mapping $\eta \in V \cap P$ with:

$$\sum\nolimits_{x \in \Upsilon_{\mathrm{A}},\, a \in \Omega_{\mathrm{A}},\, y, y' \in \Upsilon_{\mathrm{B}},\, b, b' \in \Omega_{\mathrm{B}}} \big|\eta\big((x,a),(y,b),(y',b')\big) - \nu_{\mathrm{B}}\big((x,a),(y,b),(y',b')\big)\big| \;\leq\; c \cdot k^{\delta(\frac{1}{2}+\beta)(\Delta-1)}$$

Further, by Lemma 34 we still have that a protocol run of π with overwhelming probability is either aborted or for all $x \in \Upsilon_{\mathrm{A}}$ it holds:

$$\nu_{\mathrm{B}}\big((x,\Omega_{\mathrm{A}}),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}}),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big) \;\geq\; \tilde{n}_{\mathrm{A}}(x) - k^{(\frac{1}{2}+\beta)(\Delta-1)} \;\geq\; k^{-\alpha} \cdot |\Upsilon_{\mathrm{A}}|^{-1} - k^{(\frac{1}{2}+\beta)(\Delta-1)}$$

Now, if $k^{-\alpha} \cdot |\Upsilon_{\mathrm{A}}|^{-1} - k^{(\frac{1}{2}+\beta)(\Delta-1)} > c \cdot k^{\delta(\frac{1}{2}+\beta)(\Delta-1)}$, we can by (4.1) conclude that a protocol run of π with overwhelming probability is either aborted or there exists a *cheating situation* $\eta \in \mathfrak{N}_{\mathrm{B}}^{(F)}$ with:

$$\sum\nolimits_{x \in \Upsilon_{\mathrm{A}},\, a \in \Omega_{\mathrm{A}},\, y, y' \in \Upsilon_{\mathrm{B}},\, b, b' \in \Omega_{\mathrm{B}}} \big|\eta\big((x,a),(y,b),(y',b')\big) - \nu_{\mathrm{B}}\big((x,a),(y,b),(y',b')\big)\big| \;\leq\; c \cdot k^{\delta(\frac{1}{2}+\beta)(\Delta-1)}$$

Note that w.l.o.g. $\delta \leq 1$, i.e. it suffices that $\alpha < \omega' := \delta(\frac{1}{2}+\beta)(1-\Delta)$ and hence $k^{-\alpha} \cdot |\Upsilon_{\mathrm{A}}|^{-1} > (c+1)k^{\delta(\frac{1}{2}+\beta)(\Delta-1)}$ for almost all $k \in \mathbb{N}$. Moreover, we could have chosen $\Delta < 1$, so that finally we can set $\varepsilon := \delta(\frac{1}{2}+\beta)(1-\Delta')$ with $\Delta < \Delta' < 1$. Thereby, we have that $c \cdot k^{\delta(\frac{1}{2}+\beta)(\Delta-1)} \leq k^{-\varepsilon}$ for almost all $k \in \mathbb{N}$ and it follows:

$$\sum\nolimits_{x \in \Upsilon_{\mathrm{A}},\, a \in \Omega_{\mathrm{A}},\, y, y' \in \Upsilon_{\mathrm{B}},\, b, b' \in \Omega_{\mathrm{B}}} \big|\eta\big((x,a),(y,b),(y',b')\big) - \nu_{\mathrm{B}}\big((x,a),(y,b),(y',b')\big)\big| \;\leq\; k^{-\varepsilon} \qquad \square$$

Lemma 38. *Let any $F := (\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Omega_{\mathrm{A}}, \Omega_{\mathrm{B}}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$ be given. Then, if Alice is honest, there exist some constants $\varepsilon, \varepsilon' \in \mathbb{R}_{>0}$, such that for any $\pi := \pi_F(X, Y, \alpha, \beta, \gamma)$ with $(X, Y, \alpha, \beta, \gamma) \in \Pi_F$ and $\alpha < \varepsilon'$ a protocol run of π with overwhelming probability is either aborted or there exists a cheating situation $\eta \in \mathfrak{N}_{\mathrm{B}}^{(F)}$ with the following properties:*

1. *We have that $\eta\big((x,a),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big) = \big|s_{\mathrm{A}}^{\mathsf{in}} \times s_{\mathrm{A}}^{\mathsf{out}} \times s_{\mathrm{B}}^{\mathsf{in}} \times s_{\mathrm{B}}^{\mathsf{out}}\big|_{(x,a,y,b)} \cdot k^{-1} \pm k^{-\varepsilon}$ for all $x \in \Upsilon_{\mathrm{A}}$, $a \in \Omega_{\mathrm{A}}$, $y \in \Upsilon_{\mathrm{B}}$, $b \in \Omega_{\mathrm{B}}$.*
2. *We have that $\eta|_{\mathrm{A}}(x) = \frac{1}{|X|} \pm k^{-\varepsilon}$ for all $x \in X$.*
3. *We have that $\eta|_{\mathrm{B}}^{\mathsf{fake}}(y) = \frac{1}{|Y|} \pm k^{-\varepsilon}$ for all $y \in Y$.*
4. *We have that $\eta|_{\mathrm{B}}^{\mathsf{fake}}(y) \leq k^{-\varepsilon}$ for all $y \in \Upsilon_{\mathrm{B}} \backslash Y$.*

If Bob is honest, the analog holds with $\eta \in \mathfrak{N}_{\mathrm{A}}^{(F)}$.

Proof. We just consider the case that Alice is honest; the analogous assertions for an honest Bob follow by symmetry reasons.

Let $\Delta > \frac{1}{2}$. Corollary 33 states that a protocol run of π is either aborted or for all $x \in \Upsilon_{\mathrm{A}}$, $a \in \Omega_{\mathrm{A}}, y \in \Upsilon_{\mathrm{B}}, b \in \Omega_{\mathrm{B}}$ it holds:

$$\nu_{\mathrm{B}}\big((x,a),(y,a),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})\big) \;=\; \big|s_{\mathrm{A}}^{\mathsf{in}} \times s_{\mathrm{A}}^{\mathsf{out}} \times s_{\mathrm{B}}^{\mathsf{in}} \times s_{\mathrm{B}}^{\mathsf{out}}\big|_{(x,a,y,b)} \cdot k^{-1} \pm k^{\Delta-(\frac{1}{2}+\beta)}$$

Further, by Lemma 37 we find some constants $\tilde{\varepsilon}, \varepsilon' \in \mathbb{R}_{>0}$, such that for any $\pi = \pi_F(X, Y, \alpha, \beta, \gamma)$ with $(X, Y, \alpha, \beta, \gamma) \in \Pi_F$ and $\alpha < \varepsilon'$ a protocol run of π with overwhelming probability is either aborted or we have:

$$\exists\, \eta \in \mathfrak{N}_{\mathrm{B}}^{(F)} :\; \sum\nolimits_{x \in \Upsilon_{\mathrm{A}},\, a \in \Omega_{\mathrm{A}},\, y, y' \in \Upsilon_{\mathrm{B}},\, b, b' \in \Omega_{\mathrm{B}}} \big|\eta\big((x,a),(y,b),(y',b')\big) - \nu_{\mathrm{B}}\big((x,a),(y,b),(y',b')\big)\big| \;\leq\; \tfrac{1}{k^{\tilde{\varepsilon}}}$$

Now, all we have to do is looking for some $\varepsilon > 0$, such that the four assertions of our proposition hold true for such an η. Assertion 1 directly follows by our considerations so far and the fact that we could have chosen $\Delta < \frac{1}{2} + \beta$. In particular, for all $x \in \Upsilon_\mathrm{A}, a \in \Omega_\mathrm{A}, y \in \Upsilon_\mathrm{B}, b \in \Omega_\mathrm{B}$ we have:

$$\eta((x,a),(y,b),(\Upsilon_\mathrm{B},\Omega_\mathrm{B})) \;=\; \big|s_\mathrm{A}^{\mathsf{in}} \times s_\mathrm{A}^{\mathsf{out}} \times s_\mathrm{B}^{\mathsf{in}} \times s_\mathrm{B}^{\mathsf{out}}\big|_{(x,a,y,b)} \cdot k^{-1} \pm \big(k^{\Delta-(\frac{1}{2}+\beta)} + k^{-\tilde{\varepsilon}}\big)$$

I.e., we just need that $\varepsilon < \frac{1}{2} + \beta - \Delta$ and $\varepsilon < \tilde{\varepsilon}$. The remaining three assertions follow by the observation that in the protocol step Check A (q.v. Figure 4.1) an honest Alice for all $x \in \Upsilon_\mathrm{A}$, $y \in \Upsilon_\mathrm{B}$ enforces the following inequality:

$$\big|s_\mathrm{A}^{\mathsf{in}} \times \hat{s}_\mathrm{B}^{\mathsf{in}}[\bar{K}_\mathrm{A}]\big|_{(x,y)} \;=\; k^{\frac{1}{2}+\beta} \cdot \tilde{n}_\mathrm{A}(x) \cdot \tilde{n}_\mathrm{B}(y) \pm k^{\frac{1}{4}+\beta} \cdot |\Omega_\mathrm{A} \times \Omega_\mathrm{B}|$$

By definition of ν_B (q.v. Notation 30) this expression is equivalent to the following:

$$\nu_\mathrm{B}((x,\Omega_\mathrm{A}),(y,\Omega_\mathrm{B}),(\Upsilon_\mathrm{B},\Omega_\mathrm{B})) \;=\; \tilde{n}_\mathrm{A}(x) \cdot \tilde{n}_\mathrm{B}(y) \pm k^{-\frac{1}{4}} \cdot |\Omega_\mathrm{A} \times \Omega_\mathrm{B}|$$

Thus, by construction of $\tilde{n}_\mathrm{A}$ and $\tilde{n}_\mathrm{B}$ (q.v. Figure 4.1) it follows for our η:

$$\begin{aligned}
\eta|_\mathrm{A}(x) &= \tfrac{1}{|X|} \pm \big(\tfrac{k^{-\alpha}}{|X|} - \tfrac{k^{-\alpha}}{|\Upsilon_\mathrm{A}|} + k^{-\frac{1}{4}} \cdot |\Upsilon_\mathrm{B} \times \Omega_\mathrm{A} \times \Omega_\mathrm{B}| + k^{-\varepsilon}\big) && \text{for all } x \in X \\
\eta|_\mathrm{B}^{\mathsf{fake}}(y) &= \tfrac{1}{|Y|} \pm \big(\tfrac{k^{-\alpha}}{|Y|} - \tfrac{k^{-\alpha}}{|\Upsilon_\mathrm{B}|} + k^{-\frac{1}{4}} \cdot |\Upsilon_\mathrm{A} \times \Omega_\mathrm{A} \times \Omega_\mathrm{B}| + k^{-\varepsilon}\big) && \text{for all } y \in Y \\
\eta|_\mathrm{B}^{\mathsf{fake}}(y) &\leq \tfrac{k^{-\alpha}}{|\Upsilon_\mathrm{B}|} + k^{-\frac{1}{4}} \cdot |\Upsilon_\mathrm{A} \times \Omega_\mathrm{A} \times \Omega_\mathrm{B}| + k^{-\varepsilon} && \text{for all } y \in \Upsilon_\mathrm{B} \backslash Y
\end{aligned}$$

So, we only additionally need that $\varepsilon < \alpha$ and $\varepsilon < \frac{1}{4}$ and we are done. □

4.7 Secure generation of correlated data

In this section we put things together by combining the results of Section 4.4 and Section 4.6. In particular, we show that our generic protocol scheme from Section 4.5 can always be instantiated such that no corrupted party can deviate from the prescribed input probabilities too much, and thus the generated data is non-trivially correlated. This suffices for implementation of OT as described in Section 3.2.

Notation 39 (Cheating quantum). For $F = (\Upsilon_\mathrm{A}, \Upsilon_\mathrm{B}, \Omega_\mathrm{A}, \Omega_\mathrm{B}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$, with $\dot{\mathfrak{N}}_F$ denoting the minimal spanning set of all normalized cheating situations for F (q.v. Lemma 10), we define:

$$\vartheta_F \;:=\; \min\{\eta|_\mathrm{B}^{\mathsf{fake}}(y') \mid y' \in \Upsilon_\mathrm{B},\, \eta \in \dot{\mathfrak{N}}_F : \eta|_\mathrm{B}^{\mathsf{fake}}(y') > 0\}$$

Lemma 40 (Quantizability of cheating situations). *Let $F := (\Upsilon_\mathrm{A}, \Upsilon_\mathrm{B}, \Omega_\mathrm{A}, \Omega_\mathrm{B}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$. Further, let $\eta \in \mathfrak{N}_\mathrm{B}^{(F)}$, $\omega \in \mathbb{R}_{\geq 0}$, such that $\omega < \frac{1}{|\Upsilon_\mathrm{B}|}$. Then there exists some $\eta' \in \mathfrak{N}_\mathrm{B}^{(F)}$ that fulfills the following two conditions:*

1. *For all $y' \in \Upsilon_\mathrm{B}$ we have the following implication:*

$$\eta|_\mathrm{B}^{\mathsf{fake}}(y') \leq \omega \cdot \vartheta_F \quad \Rightarrow \quad \eta'|_\mathrm{B}^{\mathsf{fake}}(y') = 0$$

2. *For all $x \in \Upsilon_\mathrm{A}$, $a \in \Omega_\mathrm{A}$, $y, y' \in \Upsilon_\mathrm{B}$, $b, b' \in \Omega_\mathrm{B}$ we have:*

$$\big|\eta((x,a),(y,b),(y',b')) - \eta'((x,a),(y,b),(y',b'))\big| \;\leq\; 2\omega \cdot |\Upsilon_\mathrm{A} \times \Upsilon_\mathrm{B}|$$

Proof. As stated in Lemma 10, the set of all normalized cheating situations for F is the convex hull of a finte set of vertices, say $\{\dot{\eta}_1, \ldots, \dot{\eta}_n\}$. Note that for all $i \in \{1, \ldots, n\}$ and all $y' \in \Upsilon_\mathrm{B}$ we have

that either $0 < \vartheta_F \leq \dot{\eta}_i|_{\mathrm{B}}^{\mathsf{fake}}(y')$ or $\dot{\eta}_i|_{\mathrm{B}}^{\mathsf{fake}}(y') = 0$ by definition of ϑ_F (q.v. Notation 39). Now, let $\tilde{\eta}$ denote the normalized version of η (cf. Corollary 9). We define:

$$Y' := \{y' \in \Upsilon_{\mathrm{B}} \mid 0 < \tilde{\eta}|_{\mathrm{B}}^{\mathsf{fake}}(y') \leq \omega \cdot \vartheta_F\}$$

W.l.o.g., we assume that $Y' \neq \emptyset$, as otherwise we could just set $\eta' := \eta$ (cf. Remark 13). Moreover, we find some $a_1, \ldots, a_n \in \mathbb{R}_{\geq 0}$, such that $\sum_{i=1}^{n} a_i \cdot \dot{\eta}_i = \tilde{\eta}$ and especially $\sum_{i=1}^{n} a_i = 1$. We define the index set $I := \{i \in \{1, \ldots, n\} \mid \dot{\eta}_i|_{\mathrm{B}}^{\mathsf{fake}}(y') > 0\}$, whereby we get:

$$\sum\nolimits_{i \in I} a_i \cdot \vartheta_F \;\leq\; \sum\nolimits_{i \in I} a_i \cdot \dot{\eta}_i|_{\mathrm{B}}^{\mathsf{fake}}(Y') \;\leq\; \tilde{\eta}|_{\mathrm{B}}^{\mathsf{fake}}(Y') \;\leq\; \omega \cdot \vartheta_F \cdot |Y'|$$

Since $\omega < \frac{1}{|\Upsilon_{\mathrm{B}}|}$ by assumption, this especially yields that $\sum_{i \in I} a_i \leq \omega \cdot |Y'| < 1$. So, we can set $J := \{1, \ldots, n\} \backslash I$ and $\tilde{\eta}' := (\sum_{i \in J} a_i)^{-1} \cdot \sum_{i \in J} a_i \cdot \dot{\eta}_i$, whereby for all $x \in \Upsilon_{\mathrm{A}}$, $a \in \Omega_{\mathrm{A}}$, $y, y' \in \Upsilon_{\mathrm{B}}$, $b, b' \in \Omega_{\mathrm{B}}$ we get:

$$\begin{aligned}
&|\tilde{\eta}((x,a),(y,b),(y',b')) - \tilde{\eta}'((x,a),(y,b),(y',b'))| \\
&= \left|\sum_{i=1}^{n} a_i \cdot \dot{\eta}_i((x,a),(y,b),(y',b')) - \frac{\sum_{i \in J} a_i \cdot \dot{\eta}_i((x,a),(y,b),(y',b'))}{\sum_{i \in J} a_i}\right| \\
&\leq \left|\sum_{i \in I} a_i \cdot \dot{\eta}_i((x,a),(y,b),(y',b'))\right| + \left|\left(1 - \frac{1}{\sum_{i \in J} a_i}\right) \cdot \sum_{i \in J} a_i \cdot \dot{\eta}_i((x,a),(y,b),(y',b'))\right| \\
&\leq \left|\sum_{i \in I} a_i\right| + \left|\left(1 - \frac{1}{\sum_{i \in J} a_i}\right) \cdot \sum_{i \in J} a_i\right| \;=\; 2\sum_{i \in I} a_i \;\leq\; 2\omega \cdot |Y'| \;\leq\; 2\omega \cdot |\Upsilon_{\mathrm{B}}|
\end{aligned}$$

Finally, we define the mapping $\eta' : (\Upsilon_{\mathrm{A}} \times \Omega_{\mathrm{A}}) \times (\Upsilon_{\mathrm{B}} \times \Omega_{\mathrm{B}})^2 \to \mathbb{R}_{\geq 0}$ by:

$$\eta'((x,a),(y,b),(y',b')) := |\Upsilon_{\mathrm{A}}| \cdot \eta|_{\mathrm{A}}(x) \cdot \tilde{\eta}'((x,a),(y,b),(y',b'))$$

Since $\tilde{\eta}'$ is normalized, by Lemma 8 it follows that $\eta' \in \mathfrak{N}_{\mathrm{B}}^{(F)}$. Now we can put things together. On the one hand, by our choice of $\tilde{\eta}$ for all $x \in \Upsilon_{\mathrm{A}}$, $a \in \Omega_{\mathrm{A}}$, $y' \in \Upsilon_{\mathrm{B}}$, $b' \in \Omega_{\mathrm{B}}$ we have (q.v. Corollary 9):

$$\frac{\eta((x,a),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}}),(y',b'))}{\eta|_{\mathrm{A}}(x)} = \frac{\tilde{\eta}((x,a),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}}),(y',b'))}{\tilde{\eta}|_{\mathrm{A}}(x)}$$

Thus, by Condition 3 of Definition 5, for all $y' \in \Upsilon_{\mathrm{B}}$ it follows:

$$\eta|_{\mathrm{B}}^{\mathsf{fake}}(y') = \tilde{\eta}|_{\mathrm{B}}^{\mathsf{fake}}(y')$$

So, for all $y' \in \Upsilon_{\mathrm{B}}$ with $\eta|_{\mathrm{B}}^{\mathsf{fake}}(y') \leq \omega \cdot \vartheta_F$ it holds that $y' \in Y'$ and hence $\eta'|_{\mathrm{B}}^{\mathsf{fake}}(y') = \tilde{\eta}'|_{\mathrm{B}}^{\mathsf{fake}}(y') = 0$ by construction. On the other hand, for all $x \in \Upsilon_{\mathrm{A}}$, $a \in \Omega_{\mathrm{A}}$, $y, y' \in \Upsilon_{\mathrm{B}}$, $b, b' \in \Omega_{\mathrm{B}}$ we can rewrite the distance $|\eta((x,a),(y,b),(y',b')) - \eta'((x,a),(y,b),(y',b'))|$ as follows:

$$|\Upsilon_{\mathrm{A}}| \cdot \underbrace{\eta|_{\mathrm{A}}(x)}_{\leq 1} \cdot \underbrace{|\tilde{\eta}((x,a),(y,b),(y',b')) - \tilde{\eta}'((x,a),(y,b),(y',b'))|}_{\leq 2\omega \cdot |\Upsilon_{\mathrm{B}}|} \qquad \square$$

Corollary 41. *Let any $F := (\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Omega_{\mathrm{A}}, \Omega_{\mathrm{B}}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$ be given. Then, if Alice is honest, there exist some constants $\varepsilon, \varepsilon' \in \mathbb{R}_{>0}$, such that for any $\pi := \pi_F(X, Y, \alpha, \beta, \gamma)$ with $(X, Y, \alpha, \beta, \gamma) \in \Pi_F$ and $\alpha < \varepsilon'$ a protocol run of π with overwhelming probability is either aborted or there exists a cheating situation $\eta \in \mathfrak{N}_{\mathrm{B}}^{(F)}$ with the following properties:*

1. *It holds that $\eta((x,a),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})) = |s_{\mathrm{A}}^{\mathsf{in}} \times s_{\mathrm{A}}^{\mathsf{out}} \times s_{\mathrm{B}}^{\mathsf{in}} \times s_{\mathrm{B}}^{\mathsf{out}}|_{(x,a,y,b)} \cdot k^{-1} \pm k^{-\varepsilon}$ for all $x \in \Upsilon_{\mathrm{A}}$, $a \in \Omega_{\mathrm{A}}$, $y \in \Upsilon_{\mathrm{B}}$, $b \in \Omega_{\mathrm{B}}$.*

2. *It holds that* $\eta|_{\mathrm{A}}(x) = \frac{1}{|X|} \pm k^{-\varepsilon}$ *for all* $x \in X$.

3. *It holds that* $\eta|_{\mathrm{B}}^{\mathsf{fake}}(y) = \frac{1}{|Y|} \pm k^{-\varepsilon}$ *for all* $y \in Y$.

4. *It holds that* $\eta|_{\mathrm{B}}^{\mathsf{fake}}(y) = 0$ *for all* $y \in \Upsilon_{\mathrm{B}} \backslash Y$.

If Bob is honest, the analog holds with $\eta \in \mathfrak{N}_{\mathrm{A}}^{(F)}$.

Proof. The proof is straightforward; we just need to combine Lemma 38 and Lemma 40. □

Lemma 42. *Let some redundancy-free* $F := (\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}, \Omega_{\mathrm{A}}, \Omega_{\mathrm{B}}, \phi) \in \mathfrak{F}_{\mathsf{fin}}$ *be given that has some OT-core. Then there also exist an OT-core* $\{(\bar{x}, \bar{a}), (\bar{x}', \bar{a}')\} \times \{(\bar{y}, \bar{b}), (\bar{y}', \bar{b}')\} \subseteq (\Upsilon_{\mathrm{A}} \times \Omega_{\mathrm{A}})^2 \times (\Upsilon_{\mathrm{B}} \times \Omega_{\mathrm{B}})^2$*, a protocol* $\pi := \pi_F(\{\bar{x}, \bar{x}'\}, \{\bar{y}, \bar{y}'\}, \alpha, \beta, \gamma)$ *with* $(\{\bar{x}, \bar{x}'\}, \{\bar{y}, \bar{y}'\}, \alpha, \beta, \gamma) \in \Pi_F$ *and a constant* $\varepsilon \in \mathbb{R}_{>0}$ *with the following property: If at least one party (Alice or Bob) is honest, a protocol run of* π *with overwhelming probability is either aborted or in the end for all* $x \in \{\bar{x}, \bar{x}'\}$, $a \in \Omega_{\mathrm{A}}$, $y \in \{\bar{y}, \bar{y}'\}$, $b \in \Omega_{\mathrm{B}}$ *it holds:*

$$\tfrac{1}{k} \cdot \left| s_{\mathrm{A}}^{\mathsf{in}} \times s_{\mathrm{A}}^{\mathsf{out}} \times s_{\mathrm{B}}^{\mathsf{in}} \times s_{\mathrm{B}}^{\mathsf{out}}[K'] \right|_{(x,a,y,b)} \;=\; \tfrac{1}{|\{\bar{x},\bar{x}'\} \times \{\bar{y},\bar{y}'\}|} \cdot \phi_{x,y}(a,b) \pm k^{-\varepsilon}$$

Proof. By assumption we have an OT-core $\{(\tilde{x}, \tilde{a}), (\tilde{x}', \tilde{a}')\} \times \{(\tilde{y}, \tilde{b}), (\tilde{y}', \tilde{b}')\} \subseteq (\Upsilon_{\mathrm{A}} \times \Omega_{\mathrm{A}})^2 \times (\Upsilon_{\mathrm{B}} \times \Omega_{\mathrm{B}})^2$. By Corollary 26 we find some $(\bar{y}, \bar{b}), (\bar{y}', \bar{b}') \in \Upsilon_{\mathrm{B}} \times \Omega_{\mathrm{B}}$, such that $\{(\tilde{x}, \tilde{a}), (\tilde{x}', \tilde{a}')\} \times \{(\bar{y}, \bar{b}), (\bar{y}', \bar{b}')\}$ also is an OT-core and every cheating situation $\eta \in \mathfrak{N}_{\mathrm{B}}^{(F)}$ with $\eta|^{\mathsf{fake}}(\{\bar{y}, \bar{y}'\}) = 1$ is equivalent to a trivial cheating situation (cf. Definition 6). Analogously, we find some $(\bar{x}, \bar{a}), (\bar{x}', \bar{a}') \in \Upsilon_{\mathrm{A}} \times \Omega_{\mathrm{a}}$, such that $\{(\bar{x}, \bar{a}), (\bar{x}', \bar{a}')\} \times \{(\bar{y}, \bar{b}), (\bar{y}', \bar{b}')\}$ is still an OT-core and for every $\eta \in \mathfrak{N}_{\mathrm{A}}^{(F)}$ with $\eta|_{\mathrm{A}}^{\mathsf{fake}}(\{\bar{x}, \bar{x}'\}) = 1$ and all $x \in \Upsilon_{\mathrm{A}}$ we also have:

$$\eta|_{\mathrm{A}}^{\mathsf{fake}}(x) \;=\; \eta|_{\mathrm{A}}^{\mathsf{true}}(x)$$

Now, let $\pi := \pi_F(\{\bar{x}, \bar{x}'\}, \{\bar{y}, \bar{y}'\}, \alpha, \beta, \gamma)$ with $(\{\bar{x}, \bar{x}'\}, \{\bar{y}, \bar{y}'\}, \alpha, \beta, \gamma) \in \Pi_F$ and let α be sufficiently small, so that we can apply Corollary 41. Henceforth, for symmetry reasons it suffices to consider the case that Alice is honest. In this case, we find by Corollary 41 some constant $\tilde{\varepsilon} \in \mathbb{R}_{>0}$, such that a protocol run of π with overwhelming probability is either aborted or there exists a cheating situation $\eta \in \mathfrak{N}_{\mathrm{B}}^{(F)}$ fulfilling the following conditions for all $x \in \Upsilon_{\mathrm{A}}$, $a \in \Omega_{\mathrm{A}}$, $y \in \Upsilon_{\mathrm{B}}$, $b \in \Omega_{\mathrm{B}}$:

$$\eta((x,a),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})) = \left| s_{\mathrm{A}}^{\mathsf{in}} \times s_{\mathrm{A}}^{\mathsf{out}} \times s_{\mathrm{B}}^{\mathsf{in}} \times s_{\mathrm{B}}^{\mathsf{out}} \right|_{(x,a,y,b)} \cdot k^{-1} \pm k^{-\tilde{\varepsilon}} \tag{4.2}$$

$$\eta|_{\mathrm{A}}(x) = \tfrac{1}{|\{\bar{x},\bar{x}'\}|} \pm k^{-\tilde{\varepsilon}} \qquad \text{if } x \in \{\bar{x}, \bar{x}'\} \tag{4.3}$$

$$\eta|_{\mathrm{B}}^{\mathsf{fake}}(y) = \tfrac{1}{|\{\bar{y},\bar{y}'\}|} \pm k^{-\tilde{\varepsilon}} \qquad \text{if } y \in \{\bar{y}, \bar{y}'\} \tag{4.4}$$

$$\eta|_{\mathrm{B}}^{\mathsf{fake}}(y) = 0 \qquad \text{if } y \in \Upsilon_{\mathrm{B}} \backslash \{\bar{y}, \bar{y}'\} \tag{4.5}$$

Note that (4.5) can be reformulated as $\eta|_{\mathrm{B}}^{\mathsf{fake}}(\{\bar{y}, \bar{y}'\}) = 1$, and thus our choice of $\bar{y}, \bar{y}'$ yields that (4.4) is equivalent to the following:

$$\eta|_{\mathrm{B}}^{\mathsf{true}}(y) = \tfrac{1}{|\{\bar{y},\bar{y}'\}|} \pm k^{-\tilde{\varepsilon}} \quad \text{for all } y \in \{\bar{y}, \bar{y}'\}$$

Hence, by (4.3) and Condition 2 of Definition 5 we have for all $x \in \{\bar{x}, \bar{x}'\}$, $a \in \Omega_{\mathrm{A}}$, $y \in \{\bar{y}, \bar{y}'\}$, $b \in \Omega_{\mathrm{B}}$:

$$\eta((x,a),(y,b),(\Upsilon_{\mathrm{B}},\Omega_{\mathrm{B}})) \;=\; \frac{\phi_{x,y}(a,b)}{|\{\bar{x},\bar{x}'\} \times \{\bar{y},\bar{y}'\}|} \pm k^{-\tilde{\varepsilon}} \left(\frac{1}{|\{\bar{x},\bar{x}'\}|} + \frac{1}{|\{\bar{y},\bar{y}'\}|} + k^{-\tilde{\varepsilon}} \right)$$

By (4.2), this yields for all $x \in \{\bar{x}, \bar{x}'\}$, $a \in \Omega_{\mathrm{A}}$, $y \in \{\bar{y}, \bar{y}'\}$, $b \in \Omega_{\mathrm{B}}$:

$$\tfrac{1}{k} \cdot \left| s_{\mathrm{A}}^{\mathsf{in}} \times s_{\mathrm{A}}^{\mathsf{out}} \times s_{\mathrm{B}}^{\mathsf{in}} \times s_{\mathrm{B}}^{\mathsf{out}} \right|_{(x,a,y,b)} \;=\; \frac{\phi_{x,y}(a,b)}{|\{\bar{x},\bar{x}'\} \times \{\bar{y},\bar{y}'\}|} \pm k^{-\tilde{\varepsilon}} \left(1 + \frac{1}{|\{\bar{x},\bar{x}'\}|} + \frac{1}{|\{\bar{y},\bar{y}'\}|} + k^{-\tilde{\varepsilon}} \right)$$

Since in the protocol step Output of π every honest party enforces that $|K'| \geq k - k^{1-\gamma}$ (q.v. Figure 4.1), we finally have for all $x \in \{\bar{x}, \bar{x}'\}$, $a \in \Omega_A$, $y \in \{\bar{y}, \bar{y}'\}$, $b \in \Omega_B$:

$$\tfrac{1}{k} \cdot \left| s_A^{\mathsf{in}} \times s_A^{\mathsf{out}} \times s_B^{\mathsf{in}} \times s_B^{\mathsf{out}}[K'] \right|_{(x,a,y,b)} = \frac{\phi_{x,y}(a,b)}{|\{\bar{x},\bar{x}'\} \times \{\bar{y},\bar{y}'\}|} \pm \left(k^{-\tilde{\varepsilon}} \left(1 + \frac{1}{|\{\bar{x},\bar{x}'\}|} + \frac{1}{|\{\bar{y},\bar{y}'\}|} + k^{-\tilde{\varepsilon}} \right) + k^{-\gamma} \right)$$

This concludes our proof, since we can choose an arbitrary constant $\varepsilon > 0$ with $\varepsilon < \min\{\tilde{\varepsilon}, \gamma\}$ and then for almost all $k \in \mathbb{N}$ estimate the error term by $k^{-\varepsilon}$. □

4.8 Conclusion of the formal basis for our completeness criteria

By Lemma 42, one can now show quite straightforwardly our final theorem. This final theorem just states that we can implement the functionality $\mathcal{F}_{\mathrm{SMCD}}^{(G,\varepsilon)}$ (q.v. Figure 3.4), instantiated such that G has some OT-core, from any redundancy-free 2-party function $F \in \mathfrak{F}_{\mathsf{fin}}$ that has some OT-core itself. Since OT can be implemented from such instantiations of $\mathcal{F}_{\mathrm{SMCD}}^{(G,\varepsilon)}$ by standard techniques (q.v. Section 3.2), this concludes the formal basis of our classification results for stateless primitives.

Theorem. *Let any redundancy-free 2-party function $F \in \mathfrak{F}_{\mathsf{fin}}$ be given that has some OT-core. Then there exist a constant $\varepsilon \in \mathbb{R}_{>0}$ and a tuple of protocol parameters $(X, Y, \alpha, \beta, \gamma) \in \Pi_F$, such that the protocol $\pi := \pi_F(X, Y, \alpha, \beta, \gamma)$ implements UC-securely the functionality $\mathcal{F}_{\mathrm{SMCD}}^{(G,\varepsilon)}$ (q.v. Figure 3.4) with some G that also has an OT-core.*

Proof. We instantiate ε and the protocol parameters $(X, Y, \alpha, \beta, \gamma)$ as needed for Lemma 42, with $X = \{\bar{x}, \bar{x}'\}$ and $Y = \{\bar{y}, \bar{y}'\}$. In particular, there exist $\bar{a}, \bar{a}' \in \Omega_A$, $\bar{b}, \bar{b}' \in \Omega_B$, such that $\{(\bar{x}, \bar{a}), (\bar{x}', \bar{a}')\} \times \{(\bar{y}, \bar{b}), (\bar{y}', \bar{b}')\}$ is an OT-core. Further, we define $G := (\Lambda_A, \Lambda_B, \psi)$ as follows:

$$\Lambda_A := \{(x,a) \in X \times \Omega_A \mid \phi_{X,Y}(a, \Omega_B) > 0\}$$
$$\Lambda_B := \{(y,b) \in Y \times \Omega_B \mid \phi_{X,Y}(\Omega_A, b) > 0\}$$
$$\psi\big((x,a),(y,b)\big) := \frac{\phi_{x,y}(a,b)}{|X \times Y|}$$

Note that G has some OT-core by construction. Furthermore, w.l.o.g. we have that $\varepsilon \leq \gamma$. Now we have to show that $\pi := \pi_F(X, Y, \alpha, \beta, \gamma)$ implements UC-securely $\mathcal{F}_{\mathrm{SMCD}}^{(G,\varepsilon)}$.

If no party is corrupted, it follows straightforwardly by Lemma 31 that π is aborted only with some negligible probability. Further, the simulator in the ideal model just has to send a compound string $t_A \times t_B$ of right length to the ideal functionality $\mathcal{F}_{\mathrm{SMCD}}^{(G,\varepsilon)}$, so that the joint output distribution of non-aborted protocol runs in the real model is identical to the joint output distribution in the ideal model. Thus, simulation in a totally uncorrupted setting is just straightforward.

If Alice (and only Alice) or Bob (and only Bob) is corrupted, we need only a slightly more sophisticated simulator program. For symmetry reasons it suffices to consider a corrupted Bob. In this case, our simulator works as follows: He lets the corrupted Bob play with a simulated version of the honest Alice and a simulated version of the hybrid functionality $\mathcal{F}_{\mathrm{SFE}}^{(F)}$, thus generating some joint output string $(s_A^{\mathsf{in}} \times s_A^{\mathsf{out}}) \times (s_B^{\mathsf{in}} \times s_B^{\mathsf{out}})[K']$ with $k - k^{1-\gamma} < |K'| \leq k$ (if the protocol is not aborted). By Lemma 42, this simulated protocol run of π with overwhelming probability is either aborted or in the end for all $x \in \{\bar{x}, \bar{x}'\}$, $a \in \Omega_A$, $y \in \{\bar{y}, \bar{y}'\}$, $b \in \Omega_B$ it holds:

$$\tfrac{1}{k} \cdot \left| s_A^{\mathsf{in}} \times s_A^{\mathsf{out}} \times s_B^{\mathsf{in}} \times s_B^{\mathsf{out}}[K'] \right|_{(x,a,y,b)} = \tfrac{1}{|\{\bar{x},\bar{x}'\} \times \{\bar{y},\bar{y}'\}|} \cdot \phi_{x,y}(a,b) \pm k^{-\varepsilon}$$

Thus, if the simulated protocol run is not aborted, the simulator can just set $t_A := s_A^{\mathsf{in}} \times s_A^{\mathsf{out}}[K']$ and $t_B := s_B^{\mathsf{in}} \times s_B^{\mathsf{out}}[K']$ and then send $t_A \times t_B$ to the ideal functionality $\mathcal{F}_{\mathrm{SMCD}}^{(G,\varepsilon)}$. Else, i.e. if the simulated protocol run is aborted, the simulator just needs to terminate, too. Again, it is straightforward to verify that the ideal model is statistically indistinguishable from the real model.

If both parties are corrupted, there is nothing to prove, since the simulator can just perfectly simulate the complete real model. □

Part II

David & Goliath
Oblivious Affine Function Evaluation

Asymptotically Optimal Building Blocks for Universally Composable Two-Party Computation from a Single Untrusted Stateful Tamper-Proof Hardware Token

5 Introduction

Recently, tamper-proof hardware tokens have received increasing attention. Tamper-proof hardware tokens allow information-theoretically secure protocols that are universally composable [Can01], they can be employed for protocols in the *globalized UC* framework [HMQU05, CDPW07], and they even allow for one-time programs, i.e. circuits that can be evaluated only once [GKR08]. However, almost all known protocols employing tamper-proof hardware are either indirect, i.e. the secure hardware is used to implement commitments or zero-knowledge proofs and additional computational assumptions must be used to obtain general two party computations [Kat07, CGS08, DNW08, MS08, DNW09], or a large number of devices must be used [GKR08, GIS$^+$10]. However, issuing multiple independent tamper-proof devices requires much stronger isolation assumptions. Not only the communication between the devices and the issuer must be prevented, but also the many devices must be mutually isolated. This is especially difficult as the devices are not necessarily trusted—e.g., see [BKMN09] for the difficulty of isolating two devices in one location.

In this work, we present a protocol that realizes universally composable two-party computations with information-theoretic security using only a single (untrusted) tamper-proof device. The main challenge when using only a single piece of tamper-proof hardware is to prevent a corrupted token from encoding previous inputs in subsequent outputs.

5.1 Related work

The idea of secure computation based on separation assumptions was introduced in [BOGKW88] to construct multi-prover interactive proof systems. In particular, [BOGKW88] proposes an unconditionally secure protocol for Rabin-OT [Rab81] between two provers and a verifier. Even though this result is not explicitly stated in the context of tamper-proof hardware[1] and is proven secure in a standalone, synchronous model, it seems reasonable that an amplified variant of the protocol of [BOGKW88] can be proven UC-secure. However, this lies beyond the scope of this thesis.

The idea of explicitly using tamper-proof hardware for cryptographic purposes was introduced by [GO96], where it was shown that tamper-proof hardware can be used for the purpose of software-protection. The interest in secure hardware and separation assumptions was renewed, when it was realized that universally secure multi-party computation can be based on the setup assumption of tamper-proof hardware tokens. The tamper-proof hardware must suffice strong separation conditions, even though a more recent result showed that the assumptions about the physical separation can be relaxed to some extent [DNW08, DNW09].

Generally, the work on secure multi-party computation with tamper-proof hardware assumptions can be divided in works dealing with either stateful or stateless hardware-tokens. In [Kat07] a scenario is considered where all parties can create and issue stateful tamper-proof hardware tokens. Using additional number-theoretic assumptions, [Kat07] implements a reusable commitment functionality in this scenario. Subsequently, [MS08] improved upon [Kat07] by constructing information-theoretically secure commitments in an asymmetric scenario, where only one out of two parties is able to issue stateful tamper-proof hardware tokens. Another improvement upon [Kat07] was by [CGS08] with stateless tokens, but still bidirectional token exchange and use of enhanced trapdoor permutations (eTDP). [HMQU05] use (stateless) signature cards, issued by a trusted authority, to achieve universal composability with respect to global setup assumptions [CDPW07].

[1] The authors of [BOGKW88] mention that the provers in their protocol might be implemented as bank-cards.

In [FPS+11] it is shown how set intersection can be computed securely using a single untrusted tamper-proof hardware token and additional computational assumptions.

[GKR08] show that using a minimalistic stateful tamper-proof hardware assumption called *one-time memory* (OTM), a new cryptographic primitive called *one-time program* (OTP) can be implemented, i.e. programs that can be evaluated exactly once. An OTM can be seen as a non-interactive version of the well-known $\binom{2}{1}$-string-OT functionality: The OTM sender stores two l-bit strings on the token and sends it to the receiver party, who can arbitrarily later choose to learn one (and only one) out of the two stored values (q.v. Figure 5.1).

Functionality $\mathcal{F}_{\mathrm{OTM}}$

Parametrized by a string length l. The variable *state* is initialized by *state* $\leftarrow$ `waiting`.

Creation:

- Upon receiving input (s_0, s_1) from the sender party Goliath, verify that *state* = `waiting` and $s_0, s_1 \in \{0,1\}^l$; else ignore that input. Next, update *state* $\leftarrow$ `sent`, record (s_0, s_1) and send (`sent`) to the adversary.
- Upon receiving a message (`Delivery`) from the adversary, verify that *state* = `sent`; else ignore that input. Next, update *state* $\leftarrow$ `delivered` and send (`ready`) to the receiver party David.

Query:

- Upon receiving input (x) from the receiver party David, verify that *state* = `delivered` and $x \in \{0,1\}$; else ignore that input. Next, update *state* $\leftarrow$ `queried` and output (s_x) to David.

When a party is corrupted, the adversary is granted unrestricted access to the channel between $\mathcal{F}_{\mathrm{OTM}}$ and the corrupted party, including the ability of deleting and/or forging arbitrary messages.

Figure 5.1: The ideal/hybrid functionality modeling a single one-time memory (OTM). Following [MS08], we call the token issuer "Goliath" and the receiver party "David"; see also Section 6.3.1.

Recently, [Kol10] implemented string-OT with stateless tamper-proof hardware tokens, but achieved only covert security [AL07]. A unified treatment of tamper-proof hardware assumptions is proposed by [GIS+10]. Important in the context of our work, they show that in a mutually mistrusting setting, trusted OTPs can be implemented statistically secure from a polynomial number of OTMs. In [GIMS10], statistically secure commitments and statistical zero-knowledge are implemented on top of a single stateless tamper-proof token. Furthermore, if tokens can be encapsulated into other tokens, general statistically secure composable multi-party computation is possible in this setting. [GIMS10] also show that unconditionally secure OT cannot be realized from stateless tamper-proof hardware alone. Finally, the latest result in this research field is by [CKS+11], that combine techniques of [GIS+10] and a preliminary version of our work [DKMQ11], resulting in a computationally secure, constant-round protocol for OT with unlimited token reusability. They only need stateless tokens and show black-box simulatability. However, this comes at the cost of bidirectional token exchange and the assumption that collision resistant hashfunctions (CRHF) exist.

Except for [BOGKW88], all of the above schemes based on untrusted tamper-proof hardware either use additional complexity assumptions to achieve secure two-party computations [HMQU05, Kat07, MS08, GKR08, DNW08, DNW09, Kol10, CKS+11] or a large number of hardware tokens must be issued [GKR08, GIS+10].

5.2 Our contribution

In this thesis we show that general, information-theoretically secure, composable two-party computations are possible in a setting where a *single* untrusted stateful tamper-proof hardware token is issued by one party. Previous solutions required that either the creator of the tamper-proof hardware is honest, that additional complexity assumptions are used, or that a large number of independent tamper-proof hardware tokens is issued. Our approach uses only a single tamper-proof token and apart from that solely relies on some linear algebra and combinatorics, what may be of independent interest. As a drawback our protocols allow only for limited token reusability. However, they can be transformed straightforwardly into computationally secure solutions with unlimited token reusability. Remarkably, for this transformation only a very weak complexity assumption is needed, namely the existence of a pseudorandom number generator (PRNG), and only the token receiver needs to be computationally bounded.

As a reasonable abstraction for the primitives that can be implemented in our setting, we introduce a new primitive, which we call *sequential one-time OAFE* (q.v. Section 6.4). We show that OT can be realized straightforwardly using this primitive; thus our results for statistically secure, composable two-party computations follow immediately by the completeness of OT [Kil88, IPS08]. At the same time, we improve upon the results of [DKMQ11] in several ways. Firstly, the computational variant of our construction allows for unlimited reuse of the tamper-proof hardware, as mentioned above, whereas in [DKMQ11] the number of token queries always was a priori bounded. Furthermore, we can still straightforwardly adapt the results of [GIS+10] to implement trusted OTPs at the cost of *one* tamper-proof hardware token per OTP (cf. Section 6.4). Last but not least, we achieve a better complexity than in [DKMQ11] (cf. Section 7.2.3 and Section 7.2.1). In particular, with our new approach one can implement several widely-used building blocks for secure multi-party computation and these constructions have some remarkable optimality features.

Sequentially queriable OTMs: We propose an information-theoretically secure construction for an arbitrary polynomial number of OTM functionalities from a single tamper-proof token. The number of OTMs must be chosen when the token is issued and cannot be increased later, unless the token contains a PRNG and the receiver is computationally bounded (i.e. we partly give up information-theoretic security). The implemented OTM instances are only queriable in a predefined order, but this can definitely be considered an advantage, since it trivially rules out the out-of-order attacks dealt with in [GIS+10]. Our construction is not truly non-interactive; it needs some interaction during an initialization phase. However, after the initialization phase no further interaction between the token receiver and the token issuer is necessary. Therefore we say that our construction is "semi-interactive". What is more, we need only two rounds of interaction, not counting for the token transmission. This is optimal for a single-token solution. Besides, our construction can be straightforwardly transformed into a truly non-interactive solution with two mutually isolated tokens. Last but not least, we achieve an asymptotically optimal communication complexity in the sense that the number of transferred bits is linear in the number and string length of the implemented OTM instances.

Admittedly, for information-theoretically secure implementation of a large number of OTMs we need that our token stores a large (though still linear) amount of data. Now, if these OTMs are used to implement a one-time program, one may ask why we do not just implement the one-time program directly on the token. There are at least three good reasons to implement an OTP via OTMs. Firstly, the token can be transferred a long time before the sender chooses which OTP to send. Secondly, via OTMs one can implement *trusted* OTPs, i.e. sender and receiver agree on a circuit to be evaluated and only the inputs for this circuit are kept secret. The crucial security feature of a trusted OTP is that even a corrupted sender cannot change the circuit. Thirdly, since our token only needs to store random values, we can dramatically compress its size at the cost of only a very weak computational assumptions, namely the

existence of a PRNG. Moreover, this computational assumption is only needed to hold for the token receiver; all our computational protocol variants are still statistically secure against a malicious token issuer and even the token may be computationally unbounded.

To sum things up, our construction has the following features:

- many OTMs (arbitrary polynomial) by a single token; upper bound fixed at initialization
- implemented OTM instances only queriable in predefined order
- optimal round complexity: two rounds using one token or one round using two tokens
- optimal communication complexity (linear in number and size of implemented OTMs)
- information-theoretic security (but large token; compression possible by PRNG)

Commitments in both directions: We also propose a constant-round construction for a bidirectional and reusable string-commitment functionality from a single tamper-proof token. We offer several protocol variants, so that one can choose between limited reusability and information-theoretic security on the one side, and unlimited reusability at the cost of computational assumptions on the other side. Anyway, for unlimited reusability we only need a PRNG and a computationally bounded token receiver; the token issuer (and even the token) may still have arbitrary computing power. What is more, by our construction one can implement an arbitrary polynomial number of commitments in parallel with $O(1)$ rounds of communication. Besides, our construction can be straightforwardly transformed into a non-interactive solution with two mutually isolated tokens, so that the whole communication of each commit and unveil phase only consists of a single message sent by the committing/unveiling party. Last but not least, we achieve an asymptotically optimal communication complexity in the sense that the number of transferred bits is linear in the number and string length of the implemented commitments. To the best of our knowledge, except for [MS08] all other constructions based on tamper-proof hardware have higher communication complexity *and* either use stronger complexity assumptions or have $\omega(1)$ rounds. However, the construction of [MS08] is only unidirectional (from the token issuer to the token receiver).

To sum things up, our construction has the following features:

- bidirectional and reusable string-commitment functionality from a single token
- unlimited reusability at the cost of a minimal complexity assumption (PRNG)
- multiple commitments with $O(1)$ rounds by one token or non-interactively by two tokens
- optimal communication complexity (linear in number and size of commitments)

String-OT: Our OT protocol enjoys the same features as our commitment protocol. We omit an explicit itemization of the features of our OT construction; it is just exactly the same as the above feature list of our commitment construction. Instead, by Figure 5.2 we compare our OT protocol with earlier results in the literature.

At this point, it is important to mention that optimal communication complexity for only computationally secure OT is no great achievement at all. The string length of *any* computationally secure OT protocol can be polynomially extended by standard techniques, what accordingly improves its efficiency: The sender party just uses the OT for transmission of two random PRNG seeds and announces the actual OT inputs one-time pad encrypted with the respective pseudorandomness. In particular, by this simple trick and some rescaling of the security parameter, one can transform any OT protocol with polynomial communication complexity into a protocol with linear (and thus optimal) communication complexity. However, we stress that nevertheless we present the first *information-theoretically* secure construction for multiple OT with optimal communication complexity based on reusable tamper-proof hardware. Moreover, note that an analogous approach for extending the string length of commitments or OTMs would destroy composability. We discuss this in further detail in Section 7.2.7.

	stateless tokens			stateful tokens (simulator needs to rewind)			
	[CGS08]	[GIS+10]	[CKS+11]	[GIS+10]	[DKMQ11]	this work	
tokens	2 (bidirect.)	$\Theta(k)$	2 (bidirect.)	$\Theta(k)$	1	1	1
rounds	$\Theta(k)$	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$
bits sent	?	$\Omega(k^2)$	$\Omega(k^2)$	$\Theta(k^2)$	$\Theta(k^2)$	$\Theta(k)$	$\Theta(k)$
assumptions	eTDP	CRHF	CRHF	none	none	none	PRNG
reusability	unbounded	none	unbounded	none	bounded	bounded	unbounded

Figure 5.2: UC-secure k-bit string-OT based on tamper-proof tokens; table partly borrowed from [CKS+11, Table 1]. The CRHF-based protocols can instead be based on one-way functions (equivalent to PRNGs), but using $\Theta(k/\log k)$ rounds. For [CGS08] an explicit estimation of the overall communication complexity is just omitted, since they use the heavy machinery of general zero-knowledge proofs, signatures, etc. However, note that the complexity of *any* computationally secure OT protocol can be amortized by standard techniques (cf. Section 7.2.7).

All our constructions also have remarkably low computation complexity, what makes them very practical. Per implemented k-bit OTM/Commitment/OT all parties and the tamper-proof token have to perform no more than $O(1)$ finite field operations (only additions and multiplications) with field size 2^k. Additionally, the protocol variants with unlimited token reusability require that the token generates $\Theta(k)$ bits of pseudorandomness respectively, but there are no exponentiations or other operations costlier than finite field multiplication.

5.3 Outline of Part II

The rest of this thesis is organized as follows. In Section 6 we introduce some notations (Section 6.1), give a short overview of the notion of security that we use (Section 6.2), describe how our tamper-proof hardware assumption is defined in that framework (Section 6.3) and introduce our new primitive (Section 6.4), which serves as the basic building block for all other constructions. In Section 7.1 we show how one can implement our new primitive from the aforementioned tamper-proof hardware assumption. In Section 7.2 we discuss refinements and some unobvious applications of our construction. At the end of Section 7.2, in Section 7.2.7, we also briefly discuss why an only computationally secure OT protocol with optimal communication complexity is not a noteworthy result, whereas the opposite is true for commitments and OTMs. In Section 8 we give a formal security proof. Finally, in Section 9 we argue for some impossibility results, give a conclusion of our work on protocols using tamper-proof hardware, and suggest directions for improvements and future research.

6 Preliminaries

6.1 Notations

Finite fields: By $\mathbb{F}_q$ we denote the finite field of size q.

Outer products: Given any field $\mathbb{F}$ and $k, l \in \mathbb{N}_{>0}$, we identify vectors in $\mathbb{F}^k$ by $(k \times 1)$-matrices, so that for all $x \in \mathbb{F}^k$ and $y \in \mathbb{F}^{1\times l}$ the matrix product $xy \in \mathbb{F}^{k\times l}$ is well-defined.

Complementary matrices: Given any field $\mathbb{F}$, some $k, l \in \mathbb{N}_{>0}$ with $k < l$ and any two matrices $C \in \mathbb{F}^{(l-k)\times l}$, $G \in \mathbb{F}^{k\times l}$, we say that G is *complementary* to C, if the matrix $M \in \mathbb{F}^{l\times l}$ generated by writing G on top of C has maximal rank in the sense that $\mathrm{rank}(M) = \mathrm{rank}(C) + k$. Note that, given any $C \in \mathbb{F}^{(l-k)\times l}$, $G \in \mathbb{F}^{k\times l}$, $x \in \mathbb{F}^l$, $y \in \mathbb{F}^k$ with G complementary to C, we can always find some $x' \in \mathbb{F}^l$, such that $Cx' = Cx$ and $Gx' = y$.

Uniform distribution: Given any random variable $\mathbf{x}$ that is uniformly random over some set X, we denote that by $\mathbf{x} \xleftarrow{\mathrm{r}} X$.

Probabilities, expected values and collision entropy: We denote the probability of a random event $\mathcal{E}$ by $\mathbb{P}[\mathcal{E}]$. The expected value of a random variable $\mathbf{x}$ is denoted by $\mathbb{E}(\mathbf{x})$ and its collision entropy by $\mathbb{H}_2(\mathbf{x}) := -\log_2\left(\sum_\alpha (\mathbb{P}[\mathbf{x} = \alpha])^2\right)$.

Statistical distance: We denote the statistical distance of two given random variables $\mathbf{x}, \mathbf{y}$ by $\mathsf{SD}(\mathbf{x}, \mathbf{y})$, using the following standard notion of statistical distance:

$$\mathsf{SD}(\mathbf{x}, \mathbf{y}) \;=\; \tfrac{1}{2}\sum\nolimits_\alpha \left|\mathbb{P}[\mathbf{x} = \alpha] - \mathbb{P}[\mathbf{y} = \alpha]\right|$$

Correlation of random variables: We define the following measure for the correlation of random variables. Given any two random variables $\mathbf{x}, \mathbf{y}$ that may depend on each other, we set $\iota(\mathbf{x}, \mathbf{y}) := \mathsf{SD}\big((\mathbf{x}, \mathbf{y}), (\tilde{\mathbf{x}}, \tilde{\mathbf{y}})\big)$ with $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$ denoting independent versions of $\mathbf{x}$ and $\mathbf{y}$ respectively. Note that $\iota(\mathbf{x}, \mathbf{y}) = 0$ if and only if $\mathbf{x}$ and $\mathbf{y}$ are statistically independent. Further note that $\mathbb{P}\left[Q(\mathbf{x}, \mathbf{y})\right] \leq \mathbb{P}\left[Q(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})\right] + \iota(\mathbf{x}, \mathbf{y})$ for every predicate Q. We will use the latter extensively.

Negligibility: A function $\mu : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is negligible (in the parameter k), if $\lim_{k\to\infty} \mu(k) \cdot f(k) = 0$ for every polynomial $f \in \mathbb{R}[X]$.

Indistinguishability: Two random variables $\mathbf{x}, \mathbf{y}$ are (statistically) indistinguishable, if their statistical distance $\mathsf{SD}(\mathbf{x}, \mathbf{y})$ is negligible in some security parameter.

6.2 Framework & notion of security

We state and prove our results in the Universal-Composability (UC) framework of [Can01]. In this framework, security is defined by comparison of an *ideal model* and a *real model*. The protocol of interest is running in the latter, where an adversary $\mathcal{A}$ coordinates the behavior of all corrupted parties. In the ideal model, which is secure by definition, an ideal functionality $\mathcal{F}$ implements the desired protocol task and a simulator $\mathcal{S}$ tries to mimic the actions of $\mathcal{A}$. An environment $\mathcal{Z}$ is plugged either to the ideal or the real model and has to guess, which model it is actually plugged

to. A protocol Π is a *universally composable* (UC-secure) implementation of an ideal functionality $\mathcal{F}$, if for every adversary $\mathcal{A}$ there exists a simulator $\mathcal{S}$, such that for all environments $\mathcal{Z}$ the entire view of $\mathcal{Z}$ in the real model (with Π and $\mathcal{A}$) is statistically close to its view in the ideal model (with $\mathcal{F}$ and $\mathcal{S}$).

In our case the adversarial entities $\mathcal{A}, \mathcal{S}$ and the environment $\mathcal{Z}$ are computationally unbounded and a hybrid functionality $\mathcal{F}_{\text{wrap}}^{\text{stateful}}$ models our tamper-proof hardware assumption (q.v. Section 6.3). If the views of $\mathcal{Z}$ in the ideal model and the real model are distributed identically, we speak of *perfect* security; if there is some negligible statistical distance between these views, we have only *statistical* security.

6.3 Modeling tamper-proof hardware

6.3.1 The hybrid functionality $\mathcal{F}_{\text{wrap}}^{\text{stateful}}$

Our formulation of general stateful tamper-proof hardware resembles the meanwhile standard definitions of [Kat07, MS08]. Following [MS08], we call the token issuer "Goliath" and the receiver party "David". This naming is also motivated by the fact that all computational versions of our protocols only need David's computing power to be polynomially bounded in the security parameter; Goliath (and even the token) may be far more powerful.

To model tamper-proof hardware, we employ the $\mathcal{F}_{\text{wrap}}^{\text{stateful}}$ wrapper functionality (q.v. Figure 6.1). The sender party Goliath provides as input a Turing machine $\mathcal{M}$ to $\mathcal{F}_{\text{wrap}}^{\text{stateful}}$. The receiver party David can now query $\mathcal{F}_{\text{wrap}}^{\text{stateful}}$ on arbitrary input words w, whereupon $\mathcal{F}_{\text{wrap}}^{\text{stateful}}$ runs $\mathcal{M}$ on input w, sends the output that $\mathcal{M}$ produced to David and stores the new state of $\mathcal{M}$. Every time David sends a new query w' to $\mathcal{F}_{\text{wrap}}^{\text{stateful}}$, it resumes simulating $\mathcal{M}$ with its most recent state, sends the output to David and updates the stored state of $\mathcal{M}$.

Functionality $\mathcal{F}_{\text{wrap}}^{\text{stateful}}$

The variable *state* is initialized by *state* $\leftarrow$ `wait`.

Creation:

- Upon receiving a message $(\texttt{Create}, \mathcal{M}, b)$ from Goliath, where $\mathcal{M}$ is the program of a deterministic interactive Turing machine and $b \in \mathbb{N}$, verify that *state* = `wait`; else ignore that input. Next, initialize a simulated version of $\mathcal{M}$, store b, set *state* $\leftarrow$ `sent` and send (`created`) to the adversary.
- Upon receiving a message (`Delivery`) from the adversary, verify that *state* = `sent`; else ignore that input. Next, set *state* $\leftarrow$ `execute` and send (`ready`) to David.

Execution:

- Upon receiving a message $(\texttt{Run}, w)$ from David, where w is an input word, verify that *state* = `execute`; else ignore that input. Next, write w on the input tape of the simulated machine $\mathcal{M}$ and carry on running $\mathcal{M}$ for at most b steps, starting from its most recent state. When $\mathcal{M}$ halts (or b steps have passed) without generating output, send a special symbol $\perp$ to David; else send the output of $\mathcal{M}$.

When a party is corrupted, the adversary is granted unrestricted access to the channel between $\mathcal{F}_{\text{wrap}}^{\text{stateful}}$ and the corrupted party, including the ability of deleting and/or forging arbitrary messages.

Figure 6.1: The wrapper functionality by which we model stateful tamper-proof hardware sent from Goliath to David. Note that delivery of the token in the creation phase is scheduled by the adversary, whereas afterwards all communication between David and the token is immediate.

This captures the following properties one expects from tamper-proof hardware. On the one hand, Goliath is unable to revoke $\mathcal{M}$ once he has sent it to David. On the other hand, David can run $\mathcal{M}$ on inputs of his choice, but the program code and state of $\mathcal{M}$ are out of reach for him, due to the token's tamper-proofness. Note that $\mathcal{M}$ does not need a trusted source of randomness, as it can be provided with a sufficiently long hard-coded random tape. Thus, w.l.o.g. we can restrict $\mathcal{M}$ to be deterministic.

For formal reasons we require that the sender party Goliath not only specifies the program code of $\mathcal{M}$, but also an explicit runtime bound $b \in \mathbb{N}$. This just ensures that even a corrupted Goliath cannot make $\mathcal{F}_{\text{wrap}}^{\text{stateful}}$ run perpetually. As we will state and prove all our results without any computational assumptions regarding the token, a corrupted Goliath may choose b arbitrarily large. However, when Goliath is honest, we will only need that the number of computing steps performed by the token is polynomial in the security parameter. We will henceforth implicitly assume that an honest Goliath always adjusts the parameter b accordingly.

6.3.2 Real world meaning of our hardware assumption and proof techniques

In Section 8 we will show that our construction from Section 7.1 is universally composable. However, the respective simulator for a corrupted sender party Goliath will need to rewind the token and thus has to know the token code. At first glance, it might seem a rather strong assumption that a corrupted token manufacturer always knows the internal program code of his tokens. How can such a party be prevented from just passing on a token received during another protocol from some uncorrupted token issuer?

We argue that tokens can be bound to the corresponding issuer IDs by not too unrealistic assumptions. The conceptually simplest (but a bit overoptimistic) way are standardized and unforgeable token cases, branded with the respective issuer ID, and that cannot be removed without destroying the token completely. However, we can go with a bit less rigorous assumptions. We just need that undetectable token encapsulation is infeasible (e.g., since the token's weight and size would be altered) and that every honestly programmed token initially outputs its manufacturer's ID. Then, only tokens of corrupted manufacturers can be successfully passed on. Since w.l.o.g. all corrupted parties collude, now every token issuer automatically knows the internal program code of all his issued and/or passed on tokens. Infeasibility of token encapsulation is also needed by [HMQU05, Kat07, MS08, GKR08].

We also argue that using a *stateful* token does not necessarily mean a categorical disadvantage compared to protocols based on *stateless* tokens. In the literature one can find the opposite point of view, usually motivated by *resetting attacks*. These attacks only affect stateful approaches, whereas stateless approaches stay secure. By a resetting attack a corrupted token receiver tries to rewind the token (e.g. by cutting off the power supply) and then run it with new input. Such an attack, if successful, would break security of all our protocols. However, as a countermeasure the tamper-proof token could delete its secrets or just switch to a special "dead state" when a resetting attempt is detected. For the technical realization we suggest, e.g., that the state information is stored as a code word of an error correcting code and the token does not work unless the stored state information is an error-free, non-trivial code word. Anyway, we consider a thorough investigation of this issue an interesting direction for future research.

6.4 Sequential one-time OAFE and its relation to OTMs and OT

There is a two-party functionality that we call *oblivious affine function evaluation* (OAFE), in the literature sometimes referred to as *oblivious linear function evaluation* (OLFE), which is closely related to OT and of particular interest for our constructions. In $\mathbb{F}_q^k$-OAFE, with q and k publicly known but not necessarily constant, the sender chooses an affine function parametrized by two vectors $a, b \in \mathbb{F}_q^k$ and the receiver chooses a preimage $x \in \mathbb{F}_q$. The receiver gets as output the

$\mathbb{F}_q^k$-vector $y := ax + b$ and the sender's output is empty. The receiver does not learn more about the sender's input (a, b) than he can infer from (x, y) and the sender does not learn anything about the receiver's input x. As one can see quite easily, $\mathbb{F}_2$-OAFE and OT can be reduced to each other without any overhead (q.v. Figure 6.2). Note that the reductions in Figure 6.2 also work perfectly for $\mathbb{F}_2^k$-OAFE and k-bit string-OT respectively.

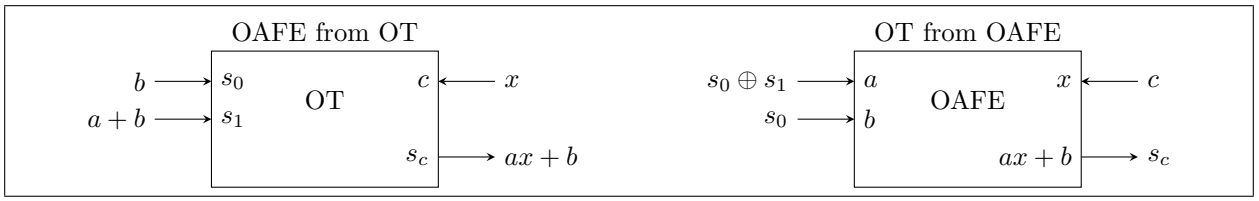

Figure 6.2: Reductions between bit-OT and $\mathbb{F}_2$-OAFE; protocols borrowed from [WW06].

We implement a variant of OAFE that we call "sequential one-time OAFE", or "seq-ot-OAFE" for short. By *one-time* OAFE we mean a primitive that works analogously to an OTM. The sender creates a token parametrized by $a, b \in \mathbb{F}_q^k$ and sends it to the receiver. Arbitrarily later the receiver may *once* input some $x \in \mathbb{F}_q$ of his choice into the token, whereupon the token outputs $y := ax + b$ and then terminates. *Sequential* one-time OAFE lets the sender send up to a polynomial number of single one-time OAFE tokens, but the receiver may only query them in the same order as they were sent. However, when the receiver has queried some of the tokens he already received, this does not vitiate the sender's ability to send some additional tokens, which in turn can be queried by the receiver afterwards, and so on. For a formal definition of the ideal seq-ot-OAFE functionality see Figure 6.3.

Functionality $\mathcal{F}_{\text{OAFE}}^{\text{seq-ot}}$

Parametrized by a finite vector space $\mathbb{F}_q^k$ and some runtime bound n that is polynomially bounded in the security parameter $\lambda := k \log q$. The counters $j_{\mathsf{created}}, j_{\mathsf{sent}}, j_{\mathsf{queried}}$ are all initialized to 0.

Send phases:

- Upon receiving input (a, b, i) from Goliath, verify that $a, b \in \mathbb{F}_q^k$ and $i = j_{\mathsf{created}} + 1 \leq n$; else ignore that input. Next, update $j_{\mathsf{created}} \leftarrow i$, record (a, b, i) and send $(\texttt{created}, i)$ to the adversary.
- Upon receiving a message $(\texttt{Delivery}, i)$ from the adversary, verify that $i = j_{\mathsf{sent}} + 1 \leq j_{\mathsf{created}}$; else ignore that message. Next, update $j_{\mathsf{sent}} \leftarrow i$ and send $(\texttt{ready}, i)$ to David.

Choice phases:

- Upon receiving input (x, i) from David, verify that $x \in \mathbb{F}_q$ and $i = j_{\mathsf{queried}} + 1 \leq j_{\mathsf{sent}}$; else ignore that input. Next, update $j_{\mathsf{queried}} \leftarrow i$ and for the recorded tuple (a, b, i) compute $y \leftarrow ax + b$ and output (y, i) to David.

When a party is corrupted, the adversary is granted unrestricted access to the channel between $\mathcal{F}_{\text{OAFE}}^{\text{seq-ot}}$ and the corrupted party, including the ability of deleting and/or forging arbitrary messages.

Figure 6.3: The ideal functionality for sequential one-time OAFE (seq-ot-OAFE). Note that send and choice phases can be executed in mixed order with the only restriction that the i-th send phase must precede the i-th choice phase. Further note that David's notifications about Goliath's inputs in the send phases are scheduled by the adversary, whereas all messages in the choice phases are delivered immediately.

Note that the reduction protocols in Figure 6.2 still can be adapted canonically to transform k-bit string-OTMs into $\mathbb{F}_2^k$-OAFE tokens and vice versa. Hence, using the seq-ot-OAFE functionality (q.v. Figure 6.3), a polynomial number of OTMs can be implemented very efficiently, but the receiver can query the single OTM tokens only in the same order as they were sent. However, the construction of [GIS+10] for trusted OTPs from OTMs still works, as there an honest receiver queries all OTM tokens in a fixed order anyway. Interestingly, the technical challenges dealt with in [GIS+10] arise from the fact that a malicious receiver might query the OTMs *out of order*. Moreover, the restriction to sequential access can be exploited to securely notify the sender that the receiver has already queried some OTM token. Therefor, every other OTM token is issued with purely random input from the sender and the receiver just announces his corresponding input-output tuple. A corrupted receiver that tries to adversarially delay his OTM queries is caught cheating with overwhelming probability, as he has only a negligible chance to correctly guess the next check announcement. Thus, we can implement a polynomial number of OT instances that are perfectly secure against the OT sender and statistically secure against the OT receiver. Still, the receiver can query the single OT instances only in the same order as they were sent, but in fact this is already premised in most protocols that build on OT. Noting that OT and OAFE can be stored and reversed [Bea96, WW06, Wul07], we conclude that in the seq-ot-OAFE hybrid model OT can be implemented in both ways (from the token sender to the token receiver and vice versa).

Finally, a remark is in place. Even though seq-ot-OAFE can be used to implement several OTPs, the sequential nature of seq-ot-OAFE demands that those OTPs can only be executed in a predefined order. If one wishes to implement several OTPs that can be evaluated in random order, as many seq-ot-OAFE functionalities have to be issued.

7 Semi-interactive seq-ot-OAFE from *one* tamper-proof token

7.1 The basic protocol

We want to implement seq-ot-OAFE (q.v. Section 6.4), using a *single* tamper-proof hardware token that is issued by one of the mutually distrusting parties. The technical challenge in doing so is twofold. Firstly, the receiver David must be able to verify that no token output does depend on any input of *previous* choice phases. Secondly, each token output must be an *affine* function of the corresponding input. However, note that the latter difficulty is only relevant if $q > 2$, as *every* function $f : \mathbb{F}_2 \to \mathbb{F}_2^k$ is affine: $f(x) = \big(f(0) + f(1)\big) \cdot x + f(0)$ for all $x \in \mathbb{F}_2$.

Our approach to solving these problems is enlarging the token's output space to dimension $(1+\alpha)k$ and letting the sender Goliath announce αk-dimensional linear hash values of the token's function parameters, which can be used by David for a consistency check; then there remains a k-dimensional part of the token's output for generation of the intended OAFE result. For technical reasons we choose $\alpha = 3$. In particular, a preliminary protocol idea can be sketched as follows:

- Goliath chooses the i-th token parameters uniformly at random, say $r, s \xleftarrow{\mathrm{r}} \mathbb{F}_q^{4k}$.
- Upon receiving the token, David announces a random check matrix $C \xleftarrow{\mathrm{r}} \mathbb{F}_q^{3k \times 4k}$.
- Goliath in turn announces $\tilde{r} := Cr$ and $\tilde{s} := Cs$.
- When David queries the token the i-th time, say he inputs some $x \in \mathbb{F}_q$ and receives output $w \in \mathbb{F}_q^{4k}$, he checks whether $Cw = \tilde{r}x + \tilde{s}$. When the check is not passed, David has caught Goliath cheating and henceforth always outputs a default value.

This way, we can implement some kind of "weak" OAFE, where the receiver additionally learns some linear projection of the sender's inputs, but by announcing $(\tilde{r}, \tilde{s})$ Goliath has committed the token to affine behavior. Otherwise, if the check would be passed for a large set of token inputs $X \subseteq \mathbb{F}_q$ and there do not exist any $r, s \in \mathbb{F}_q^{4k}$ such that $\tau(x) = rx + s$ for all $x \in X$ with τ denoting the token functionality in the i-th round, then the token could as well form collisions for the universal hash function C, of which it is oblivious. Moreover, we can nullify the receivers additional knowledge about (r, s) by multiplication with any matrix $G \in \mathbb{F}_q^{k \times 4k}$ that is complementary to C. When David just outputs Gw, we have implemented OAFE with random input (Gr, Gs) from Goliath and arbitrarily selectable input x from David. Finally, Goliath can derandomize his input to arbitrarily selectable $a, b \in \mathbb{F}_q^k$ by announcing $\tilde{a} := a - Gr$ and $\tilde{b} := b - Gs$. David then just has to replace his output by $y := Gw + \tilde{a}x + \tilde{b}$.

However, there is still a security hole left, as the token might act honestly only on some specific input set $X \subsetneq \mathbb{F}_q$ or even only on some specific type of *input history*. Now, when David's inputs match this adversarially chosen specification, he will produce regular output; else a protocol abortion is caused with overwhelming probability (i.e. David produces default output). Such a behavior cannot be simulated in the ideal model, unless the simulator gathers some information about David's input. Thus, David must keep his real input x secret from the token (and as well from Goliath, of course). However, David's input must be reconstructible from the *joint* view of Goliath and the token, as otherwise a corrupted David could evaluate the function specified by Goliath's input (a, b) on more than one input x. Our way out of this dilemma is by a linear secret

sharing scheme, whereby David shares his input x between Goliath and the token. In particular, the protocol now roughly proceeds as follows:

- Goliath initializes the token with uniformly random parameters $r \xleftarrow{r} \mathbb{F}_q^{4k}$ and $S \xleftarrow{r} \mathbb{F}_q^{4k\times k}$.
- Upon receiving the token, David announces a random check matrix $C \xleftarrow{r} \mathbb{F}_q^{3k\times 4k}$ and a random share $h \xleftarrow{r} \mathbb{F}_q^k \setminus \{0\}$. David and Goliath also agree on some $G \in \mathbb{F}_q^{k\times 4k}$ complementary to C.
- Goliath announces the check information $\tilde{r} := Cr$ and $\tilde{S} := CS$ and the derandomization information $\tilde{a} := a - Gr$ and $\tilde{b} := b - GSh$, where $(a, b) \in \mathbb{F}_q^k \times \mathbb{F}_q^k$ is his OAFE input.
- David randomly picks a second share $z \xleftarrow{r} \{\tilde{z} \in \mathbb{F}_q^{1\times k} \mid \tilde{z}h = x\}$, where $x \in \mathbb{F}_q$ is his OAFE input. He inputs z into the token, whereupon the token has to compute and output $W := rz + S$. When the check $CW \stackrel{?}{=} \tilde{r}z + \tilde{S}$ is passed, David computes and outputs $y := GWh + \tilde{a}x + \tilde{b}$; else he outputs some default value.

Now, neither Goliath nor the token can gather non-negligible information about David's OAFE input x. Given any set of token inputs $Z \subseteq \mathbb{F}_q^{1\times k}$ adversarially chosen in advance, the hyperplanes $\{\tilde{z} \in \mathbb{F}_q^{1\times k} \mid zh = x\}_{x\in\mathbb{F}_q}$ will partition Z into q subsets of roughly equal size, since h is uniformly random. In other words, when the token behaves dishonestly on some input set $Z \subsetneq \mathbb{F}_q^{1\times k}$, the abort probability is practically independent of David's input x.

A remarkable property of our protocol is that David's input x is only needed in the last step, where no further communication with Goliath takes place. So, we can partition the protocol into an interactive phase (where Goliath provides his OAFE input) and a non-interactive phase (where David provides his input and learns his output). Therefore, we say that our protocol is "semi-interactive". A formal description of the full protocol $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$ is given in Figure 7.1.

There are two crucial differences between $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$ and the construction in [DKMQ11]. Firstly, we changed from $\mathbb{F}_2$ to $\mathbb{F}_q$ with the explicit option that q may depend on the security parameter. This will enable us to implement OTMs, string-OT and string-commitments at optimal communication rate (cf. Section 7.2.1 and Section 7.2.5). Secondly, due to a new security proof we no longer need that Goliath's "commitments" $(\tilde{r}_1, \tilde{S}_1), \ldots, (\tilde{r}_n, \tilde{S}_n)$ are statistically independent of David's input shares $h_1, \ldots, h_n$. This allows for multiple send phases and choice phases in mixed order, so that a token that shares some random source with its issuer Goliath can be reused over and over again without any predefined limit (cf. Section 7.2.4).

At this point we also want to point out that in the protocol description of Figure 7.1 we purposely do not exactly specify how the parameters k and q depend on the security parameter λ. In fact, for our security proof we only need that $k \cdot \log q = \lambda$ and $k \geq 5$; e.g. one can choose k to be constant and q to increase exponentially. With parameters chosen this way, our protocol $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$ has only linear communication complexity, what is clearly optimal. The condition that $k \geq 5$ results from our proof techniques and is probably not tight. If $k = 1$, the protocol is not UC-secure against a corrupted sender party (see Remark 43 below), but for $2 \leq k \leq 4$ we are not aware of any potential attack. However, note that $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$ with $k < 5$ can be implemented from $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$ with $k = 5$ straightforwardly and the reduction protocol itself has only linear overhead. Thus, the asymptotic optimality of our construction with $k = 5$ does directly carry over to the case that $k < 5$.

Finally, we want to note that our protocol allows any polynomial number of send phases to be performed in parallel, so that one can still issue the polynomially many OTMs needed for an OTP by just constantly many rounds of communication (cf. Section 7.2.3).

Remark 43. Our protocol $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$ is not UC-secure against a corrupted sender Goliath, if $k = 1$.

Proof. The problem with $k = 1$ basically arises from the fact that in this case Goliath's shares h_i of David's inputs x_i are invertible field elements. Consider a maliciously programmed token that stops functioning after the first choice phase, if $z_1 \in Z$ for some adversarially chosen $Z \subseteq \mathbb{F}_q$, e.g.

with $|Z| = \frac{q}{2}$, and otherwise just follows the protocol. Since Goliath knows Z and learns h_1 during the protocol, he also knows exactly on which inputs x_1 the token breaks: It breaks, if $x_1 h_1^{-1} \in Z$. In other words, it depends on x_1, if David's outputs $y_2, \ldots, y_n$ are all-zero or not. This is not simulatable in the ideal model, because the simulator gets absolutely no information about the uncorrupted David's inputs. □

Protocol $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$

Parametrized by a finite vector space $\mathbb{F}_q^k$ and some runtime bound n that is polynomially bounded in the security parameter $\lambda := k \log q$. The setup phase is executed right at the start of the first send phase.

Setup phase:

i. For $i = 1, \ldots, n$, Goliath chooses a random vector $r_i \xleftarrow{\mathrm{r}} \mathbb{F}_q^{4k}$ and a random matrix $S_i \xleftarrow{\mathrm{r}} \mathbb{F}_q^{4k \times k}$, creates a token $\mathcal{T}$ with parameters $(r_1, S_1), \ldots, (r_n, S_n)$ and sends $\mathcal{T}$ to David via $\mathcal{F}_{\mathrm{wrap}}^{\mathrm{stateful}}$. The token also contains a counter j'_{queried} and Goliath has a counter j_{created}, both initialized to 0.

ii. Having received $\mathcal{T}$, David chooses a random matrix $C \xleftarrow{\mathrm{r}} \mathbb{F}_q^{3k \times 4k}$, computes some $G \in \mathbb{F}_q^{k \times 4k}$ complementary to C and sends (C, G) to Goliath. Furthermore, David initializes two counters $j_{\mathsf{queried}}, j_{\mathsf{sent}} \leftarrow 0$ and an initial flag $f_0 \leftarrow \top$.

iii. If Goliath finds G not complementary to C, he aborts the protocol.

Send phases:

1. Upon input (a_i, b_i, i) from the environment, Goliath verifies that $a_i, b_i \in \mathbb{F}_q^k$ and $i = j_{\mathsf{created}} + 1 \leq n$; else he ignores that input. Next, Goliath updates $j_{\mathsf{created}} \leftarrow i$, computes $\tilde{r}_i \leftarrow C r_i$ and $\tilde{S}_i \leftarrow C S_i$ and sends $(\tilde{r}_i, \tilde{S}_i, i)$ to David.
2. David chooses a random vector $h_i \xleftarrow{\mathrm{r}} \mathbb{F}_q^k \setminus \{0\}$ and sends (h_i, i) to Goliath.
3. Goliath computes $\tilde{a}_i \leftarrow a_i - G r_i$ and $\tilde{b}_i \leftarrow b_i - G S_i h_i$ and sends $(\tilde{a}_i, \tilde{b}_i, i)$ to David, who ignores that message if not $i = j_{\mathsf{sent}} + 1 \leq n$.
4. David updates $j_{\mathsf{sent}} \leftarrow i$ and outputs (ready, i) to the environment.

Throughout the whole send phase, obviously malformed messages are just ignored by the respective receiver.

Choice phases:

5. Upon input (x_i, i) from the environment, David verifies that $x_i \in \mathbb{F}_q$ and $i = j_{\mathsf{queried}} + 1 \leq j_{\mathsf{sent}}$; else he ignores that input. Next, he updates $j_{\mathsf{queried}} \leftarrow i$, chooses a random vector $z_i \xleftarrow{\mathrm{r}} \{\tilde{z} \in \mathbb{F}_q^{1 \times k} \mid \tilde{z} h_i = x_i\}$ and inputs (z_i, i) into the token $\mathcal{T}$.
6. The token verifies that $z_i \in \mathbb{F}_q^{1 \times k}$ and $i = j'_{\mathsf{queried}} + 1 \leq n$; else it ignores that input. Next, the token updates $j'_{\mathsf{queried}} \leftarrow i$, computes $W_i \leftarrow r_i z_i + S_i$ and outputs W_i to David.
7. David verifies that $f_{i-1} = \top$ and $C W_i = \tilde{r}_i z_i + \tilde{S}_i$; if $W_i \notin \mathbb{F}_q^{4l \times k}$, it is treated as an encoding of the all-zero matrix in $\mathbb{F}_q^{4l \times k}$. If the check is passed, David sets $f_i \leftarrow \top$ and computes $y_i \leftarrow G W_i h_i + \tilde{a}_i x_i + \tilde{b}_i$; otherwise he sets $f_i \leftarrow \bot$ and $y_i \leftarrow 0$ (such that $y_i \in \mathbb{F}_q^k$). Then he outputs (y_i, i) to the environment.

Figure 7.1: A protocol for semi-interactive sequential OAFE, using *one* tamper-proof token. Note that several send and choice phases can be executed in mixed order with the only restriction that an honest David will not enter the i-th choice phase before the i-th send phase has been completed.

7.2 Refinements and applications of our construction

Before we give a formal security proof for our protocol $\Pi_{\text{OAFE}}^{\text{semi-int}}$, we first want to present how the claimed optimal constructions for multiple OTMs, Commitments and OT (cf. Section 5.2) do work.

As mentioned above, we will prove security of our protocol $\Pi_{\text{OAFE}}^{\text{semi-int}}$ only for the case that $k \geq 5$. However, $\mathcal{F}_{\text{OAFE}}^{\text{seq-ot}}$ with $k < 5$ can be implemented from $\mathcal{F}_{\text{OAFE}}^{\text{seq-ot}}$ with $k = 5$ straightforwardly and the reduction protocol itself has only linear overhead. Thus, the asymptotic optimality of our construction for $\mathcal{F}_{\text{OAFE}}^{\text{seq-ot}}$ with $k = 5$ does directly carry over to the case that $k < 5$.

At the end of this section, in Section 7.2.7, we also discuss why for computationally secure OT protocols an improvement of the communication complexity is not a noteworthy result. However, this does neither affect statistically secure OT nor any commitment or OTM constructions.

7.2.1 Unidirectional string-OT and OTMs with optimal communication complexity

As discussed in Section 6.4, one can reduce k-bit string-OT and $\mathbb{F}_2^k$-OAFE to each other without any overhead. However, our construction for seq-ot-OAFE has communication complexity $\Theta(nk^2 \log q)$. I.e., by the aforementioned reduction approach we would end up with a communication complexity of $\Theta(k^2)$ per implemented instance of k-bit string-OT, as it happened in [DKMQ11]. In contrast, if k is constant and q grows exponentially in the security parameter, we have only a communication complexity of $O(\log q)$ for each implemented instance of $\mathbb{F}_q^k$-OAFE (q.v. Figure 7.1), what is clearly optimal. Therefore, it is desirable to implement l-bit string-OT by a constant number of $\mathbb{F}_{2^l}^d$-OAFE instances with constant dimension d. We present such a reduction protocol in Figure 7.2; our construction needs only a single instance of $\mathbb{F}_{2^l}^2$-OAFE and the protocol idea is as follows. The $\mathbb{F}_{2^l}^2$-OAFE primitive allows the sender party to specify two affine functions $f_0, f_1 : \mathbb{F}_{2^l} \to \mathbb{F}_{2^l}$, such that the receiver party can evaluate both functions only once and only simultaneously on the same input. Thus, if the sender party announces its OT-inputs s_0 and s_1 encrypted with $f_0(0)$ and $f_1(1)$ respectively, then the receiver party may learn at most one of the values needed for decryption of s_0 and s_1. One can even go without transmitting any ciphertexts: The sender party just has to choose f_0, f_1, such that $f_0(0) = s_0$ and $f_1(1) = s_1$, whereas $f_0(1)$ and $f_1(0)$ are completely random.

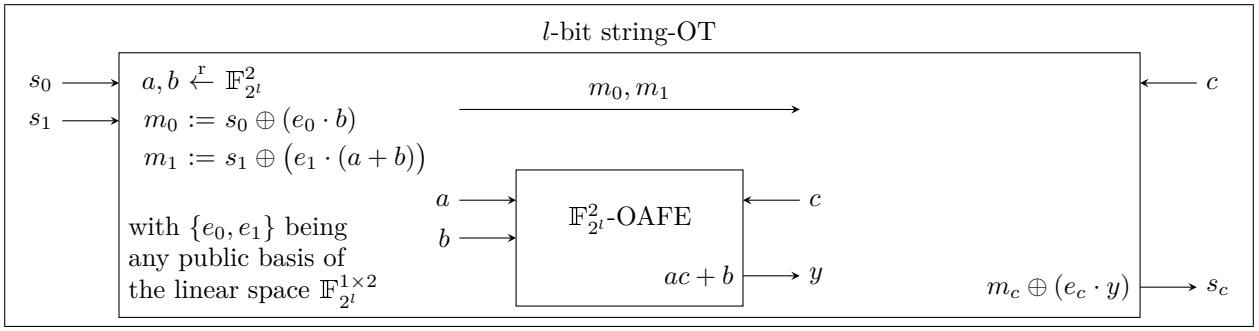

Figure 7.2: Reduction of l-bit string-OT to $\mathbb{F}_{2^l}^2$-OAFE. Note that the transmission of m_0 and m_1 is not essential; instead the sender party can just choose (a, b) subject to the condition that $e_0 \cdot b = s_0$ and $e_1 \cdot (a + b) = s_1$.

The protocol in Figure 7.2 is perfectly UC-secure, what can be shown straightforwardly, and it also works perfectly for implementation of sequentially queriable OTM tokens from seq-ot-OAFE (cf. the respective discussion in Section 6.4). Thus, in the outcome we also have a construction for sequentially queriable $\log(q)$-bit OTM tokens, using only $\Theta(\log q)$ bits of communication per implemented OTM token. This communication complexity is clearly optimal and to the best of our knowledge our approach is the first to implement statistically secure OT (or OTMs respectively) with optimal communication complexity, while based only on untrusted tamper-proof hardware.

Note that our protocols with linear communication complexity also have very low computation complexity. Per implemented $\log(q)$-bit string-OT (or $\log(q)$-bit OTM respectively) every party (and in particular the exchanged token) has only to perform $O(1)$ finite field operations with field size q, what is considerably faster than, e.g., something based on modular exponentiation.

7.2.2 Achieving optimal communication complexity for bidirectional string-OT

In Section 7.2.1 we have shown how one can implement unidirectional string-OT (from the token issuer to the token receiver) with optimal communication complexity, using our protocol $\Pi_{\text{OAFE}}^{\text{semi-int}}$ as a building block. Implementing string-OT in the other direction (from the token receiver to the token issuer) with optimal communication complexity turns out a bit more challenging. The starting point for our construction is the protocol in Figure 7.3 for reversing the direction of a given $\mathbb{F}_q$-OAFE primitive. Note that this protocol is *not* UC-secure, since a corrupted sender can cause the receiver to output some y before a and b are fixed: The corrupted sender can just send a random $m \in \mathbb{F}_q$ and arbitrarily later input some $a \in \mathbb{F}_q$ of his choice into the underlying $\mathbb{F}_q$-OAFE instance (and then compute $b := m - z$). This breaches UC-security, since an ideal version of the reversed $\mathbb{F}_q$-OAFE primitive would not send y to the receiver party before the sender's inputs a and b are fixed. However, in our case this problem has a simple solution: Since our protocol $\Pi_{\text{OAFE}}^{\text{semi-int}}$ implements sequentially queriable OAFE instances, it suffices to use every other OAFE instance for a check announcement, i.e. both parties just input randomness and the receiver has to announce his input-output tuple (cf. Section 6.4).

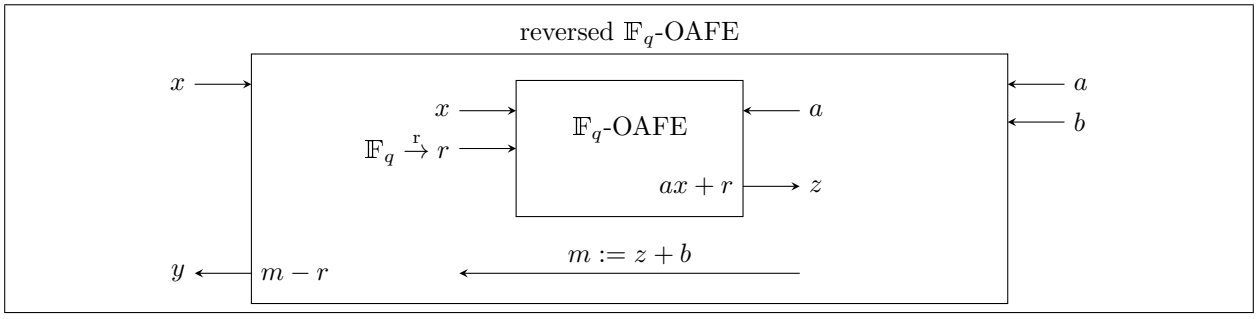

Figure 7.3: Basic approach for reversing the direction of a given $\mathbb{F}_q$-OAFE primitive; protocol taken from [WW06]. Note that this protocol is not UC-secure, unless input of a into the underlying $\mathbb{F}_q$-OAFE instance is enforced before the receiver outputs y; otherwise a corrupted sender can maliciously delay his choice of a (and b).

Obviously, the approach in Figure 7.3 does not work for $\mathbb{F}_q^k$-OAFE with $k > 1$, but we need $\mathbb{F}_q^2$-OAFE for our aimed at OT protocol. Thus, a construction for $\mathbb{F}_q^k$-OAFE from some instances of $\mathbb{F}_q$-OAFE would come in very handy. In [DKMQ12b] one can find such a construction and a security proof for the case that $k \log q$ increases polynomially in the security parameter. For the sake of self-containedness we recap in Figure 7.4 the approach of [DKMQ12b] with $k = 2$. By combining this with the protocol in Figure 7.3 and some optimization in the number of $\mathbb{F}_q$-OAFE instances used for check announcements we end up with the protocol depicted in Figure 7.5.

Now, by plugging the protocol of Figure 7.5 on top of $\Pi_{\text{OAFE}}^{\text{semi-int}}$ (q.v. Figure 7.1) we get sequentially queriable $\mathbb{F}_q^2$-OAFE from the token receiver to the token sender with an overall communication complexity of $O(\log q)$ per implemented $\mathbb{F}_q^2$-OAFE instance. So finally, we can apply again the protocol of Figure 7.2 and thereby implement string-OT with optimal communication complexity also in the direction from the token receiver to the token issuer.

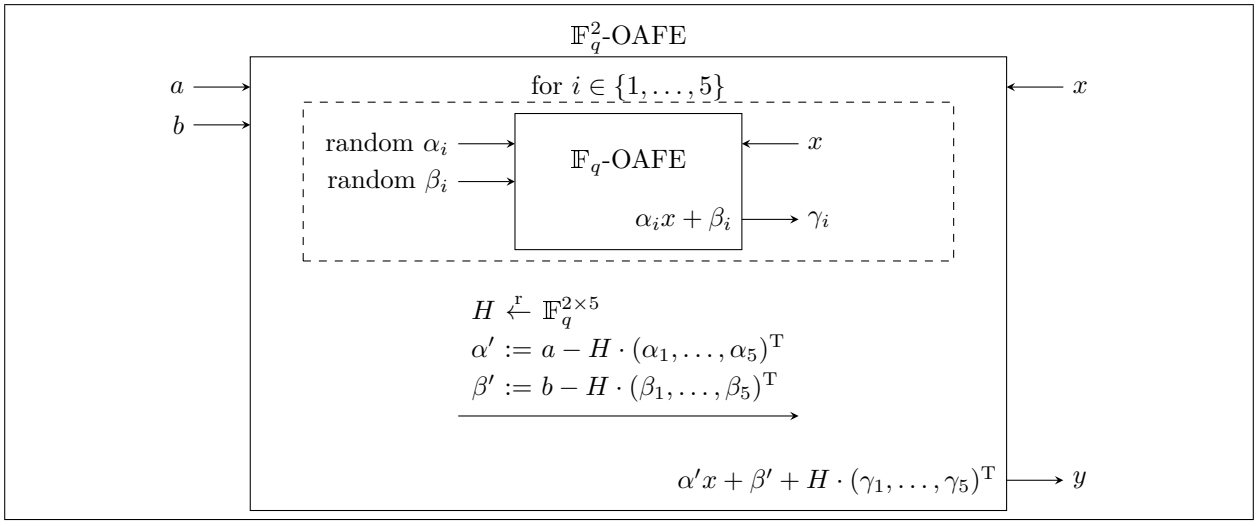

Figure 7.4: Implementation of $\mathbb{F}_q^2$-OAFE from five instances of $\mathbb{F}_q$-OAFE; protocol taken from [DKMQ12b]. Additional measures must be taken so that H is not announced before the receiver party has provided some input to all five underlying $\mathbb{F}_q$-OAFE instances in the dashed box; otherwise the protocol is inherently insecure, as shown in [DKMQ12b, Lemma 1].

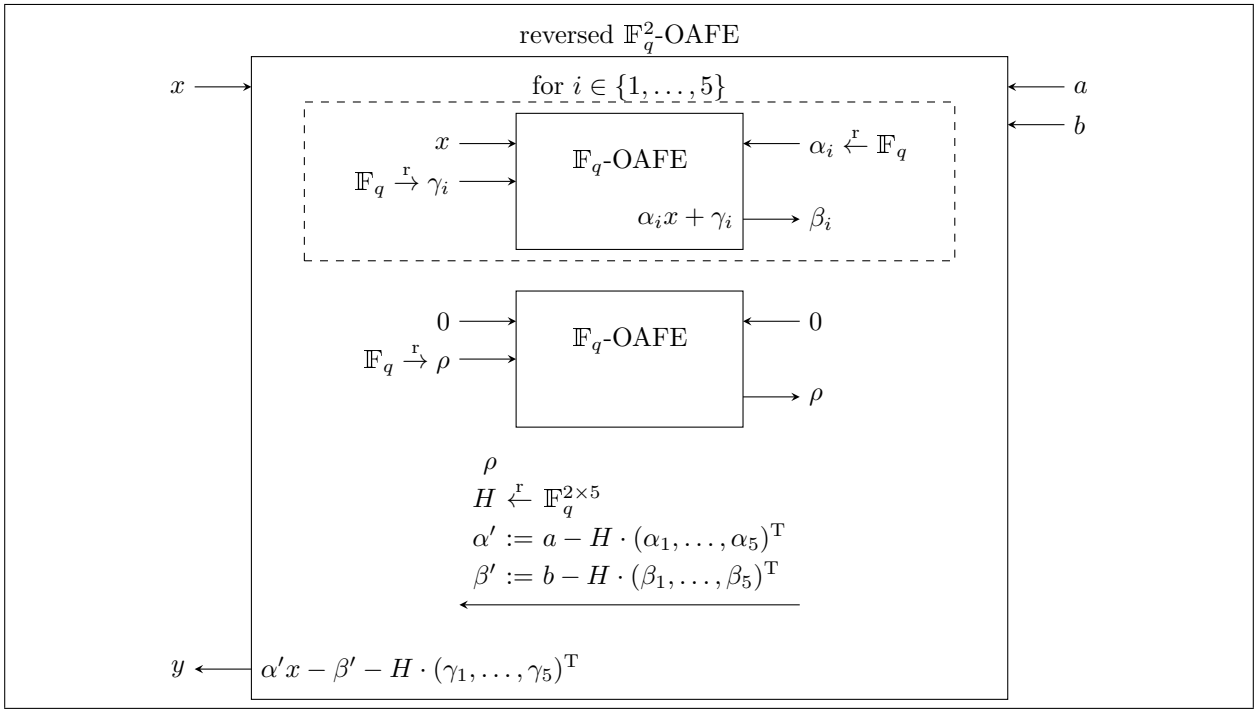

Figure 7.5: Combined protocol for UC-secure reversed $\mathbb{F}_q^2$-OAFE from six sequentially queriable instances of $\mathbb{F}_q$-OAFE. Note that the receiver must not output y unless ρ was announced correctly by the sender party.

7.2.3 Reducing the number of rounds, e.g. for one-time programs

In [GIS+10] so-called *trusted OTPs* are implemented from a polynomial amount of OTM tokens. As an honest receiver will query these tokens in some predefined (and publicly known) order, we can adapt the results of [GIS+10] to implement trusted OTPs from a single untrusted hardware token (cf. Section 6.4). However, if one implements some polynomial number (say l) of sequentially queriable OTM tokens by the construction we proposed in Section 6.4, one will end up with more than thrice as many (i.e. $3l$) rounds of communication between David and Goliath. This round complexity can be dramatically reduced as follows: In our protocol $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$ (q.v. Figure 7.1), instead of performing a large number of individual send phases, David can already announce $h_1, \ldots, h_l$ along with the check matrix C in step ii of the setup phase and Goliath can send all his announcements of the corresponding l send phases in one single message $((\tilde{r}_1, \tilde{S}_1, 1), (\tilde{a}_1, \tilde{b}_1, 1), \ldots, (\tilde{r}_l, \tilde{S}_l, l), (\tilde{a}_l, \tilde{b}_l, l))$. Thereby we end up with two rounds of communication, not counting for the transmission of the token. This modification of the protocol $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$ does not breach its security: In our formal security proof we even assume that a corrupted Goliath's announcement of $((\tilde{r}_1, \tilde{S}_1), \ldots, (\tilde{r}_n, \tilde{S}_n))$ may arbitrarily depend on $(h_1, \ldots, h_n)$. Hence, our security proof does directly carry over to the modified protocol. Note that analogously we just can arbitrarily parallelize multiple send phases of our protocol $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$ without jeopardizing security. This can be used, e.g., to implement polynomially many OTs (cf. Section 7.2.1) or commitments (cf. Section 7.2.5) with constant round complexity.

It is quite straightforward to see that a two-round protocol for implementation of polynomially many OTM tokens from a single piece of untrusted tamper-proof hardware is optimal—cf. [DKMQ11, Theorem 1]. Furthermore, our new two-round protocol is an improvement upon [DKMQ11], where we needed four rounds of communication between David and Goliath.

7.2.4 Computational solution for unlimited token reusability

Our protocol $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$ (q.v. Figure 7.1) guarantees perfect security against David (cf. Section 8.2). However, to achieve this, the token needs to be able to store $\Theta(nk^2 \log q)$ bits of information. This contradicts the idea of a tamper-proof hardware token being a small and simple device. In [MS08] it was noted, that if David is computationally bounded, then the functions stored on the token could be chosen to be pseudorandom [GGM86, HILL99]. The same is true for our construction. It suffices that the token stores a succinct seed of length $\Theta(k \log q)$ for a pseudorandom number generator F. Upon input (z_i, i) the token can compute the next pseudorandom value $(r_i, S_i) = F(i)$ and output $W_i = r_i z_i + S_i$.

Moreover, in such a setting we do not need our protocol $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$ and the ideal functionality $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$ to be parametrized by an explicit runtime bound n, as David's computational boundedness implies a polynomial upper bound for the number of token queries.

7.2.5 Efficient protocol for string-commitments in any direction

At this point we also want to note that string commitments can be implemented directly from seq-ot-OAFE, even if the dimension k is constant (i.e. q grows exponentially in the security parameter). See Figure 7.6 for the reduction protocols; they work analogously to the standard constructions for commitments from OT. As our protocol for seq-ot-OAFE with constant dimension k has only linear complexity, we thus get asymptotically optimal protocols for string commitments.

Protocol $\Pi_{\mathrm{COM}}^{\mathrm{forward}}$ (Goliath is the committing/unveiling party)

Parametrized by a string length l, which also serves as security parameter, and some runtime bound n that is polynomially bounded in l. All parties have access to a hybrid functionality $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$ parametrized by the finite vector space $\mathbb{F}_{2^l}^1$ and with runtime bound n. Bit strings of length l and elements of $\mathbb{F}_{2^l}$ are identified with each other. The counter j, held by Goliath, is initialized to 0.

Commit phases:

1. Upon input $(\texttt{Commit}, s_i, i)$ from the environment, Goliath verifies that $s_i \in \{0,1\}^l$ and $i = j+1 \leq n$; else he ignores that input. Next, Goliath updates $j \leftarrow i$, chooses some random $b_i \xleftarrow{\mathrm{r}} \mathbb{F}_{2^l}$ and sends (s_i, b_i, i) to $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$.
2. David, upon receiving the message $(\texttt{ready}, i)$ from $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$, picks some random $x_i \xleftarrow{\mathrm{r}} \mathbb{F}_{2^l}$. He sends (x_i, i) to $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$, receives some (y_i, i) and outputs $(\texttt{committed}, i)$.

Unveil phases:

3. Upon input $(\texttt{Unveil}, i)$ from the environment, Goliath verifies that $i \leq j$; else he ignores that input. Next, Goliath sends (s_i, b_i, i) to David.
4. David verifies that $s_i x_i + b_i = y_i$. If the check is passed, he outputs (s_i, i); otherwise he outputs $(\perp, i)$.

Protocol $\Pi_{\mathrm{COM}}^{\mathrm{backward}}$ (David is the committing/unveiling party)

Parametrized by a string length l, which also serves as security parameter, and some runtime bound n that is polynomially bounded in l. All parties have access to a hybrid functionality $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$ parametrized by the finite vector space $\mathbb{F}_{2^l}^1$ and with runtime bound $2n$. Bit strings of length l and elements of $\mathbb{F}_{2^l}$ are identified with each other. The counter j, held by David, is initialized to 0.

Commit phases:

1. Upon input $(\texttt{Commit}, s_i, i)$ from the environment, David verifies that $s_i \in \{0,1\}^l$ and $i = j+1 \leq n$; else he ignores that input. Next, David updates $j \leftarrow i$ and sends (i) to Goliath.
2. Goliath randomly picks $a_i, b_i, c_i, d_i \xleftarrow{\mathrm{r}} \mathbb{F}_{2^l}$ and sends $(a_i, b_i, 2i-1)$ and $(c_i, d_i, 2i)$ to $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$.
3. David, after receiving the messages $(\texttt{ready}, 2i-1)$ and $(\texttt{ready}, 2i)$ from $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$, sends $(s_i, 2i-1)$ and $(0, 2i)$ to $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$. He receives some $(y_i, 2i-1)$ and $(r_i, 2i)$ and announces (r_i, i) to Goliath.
4. Goliath outputs $(\texttt{committed}, i)$.

Unveil phases:

5. Upon input $(\texttt{Unveil}, i)$ from the environment, David verifies that $i \leq j$; else he ignores that input. Next, David sends (s_i, y_i, i) to Goliath.
6. Goliath verifies that $r_i = d_i$ and $y_i = a_i s_i + b_i$. If the check is passed, he outputs (s_i, i); otherwise he outputs $(\perp, i)$.

Figure 7.6: Asymptotically optimal protocols for string commitments from seq-ot-OAFE. Note that in a straightforward manner one can use the same instance of $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$ for both protocols simultaneously.

7.2.6 Non-interactive solution with two tokens

Our approach still needs the receiver party David to send some messages to the sender party Goliath. In particular, for each implemented instance of $\mathbb{F}_q^k$-OAFE we have an interactive send phase and a non-interactive choice phase (q.v. Figure 7.1). Therefore, we say that our protocol $\Pi_{\text{OAFE}}^{\text{semi-int}}$ is "semi-interactive". It is quite straightforward to see that one cannot implement $\mathcal{F}_{\text{OAFE}}^{\text{seq-ot}}$ from a single instance of $\mathcal{F}_{\text{wrap}}^{\text{stateful}}$ by any non-interactive protocol—cf. [DKMQ11, Theorem 1]. However, we can easily give a generic non-interactive protocol for $\mathcal{F}_{\text{OAFE}}^{\text{seq-ot}}$, if *two* instances of $\mathcal{F}_{\text{wrap}}^{\text{stateful}}$ are in place, i.e. the sender party Goliath issues two tamper-proof tokens and the receiver party David can trust that the tokens are mutually isolated. Then, the second token can play Goliath's role in the protocol $\Pi_{\text{OAFE}}^{\text{semi-int}}$ with random inputs a_i and b_i. As Goliath knows the second token's random coins, derandomization of his inputs can be done as follows: If Goliath wants to replace the random input tuple (a_i, b_i) by some arbitrarily chosen (a_i', b_i'), he just sends $(a_i' - a_i, b_i' - b_i, i)$ to David, who then has to replace his output y_i by $y_i' := y_i + (a_i' - a_i)x_i + (b_i' - b_i)$.

Note that based on the two-token protocol that implements $\mathcal{F}_{\text{OAFE}}^{\text{seq-ot}}$ with random Goliath inputs, step 2 of $\Pi_{\text{COM}}^{\text{backward}}$ (q.v. Figure 7.6) can be made non-interactive, as Goliath does not need to derandomize any of his inputs. All other protocols become non-interactive straightforwardly.

7.2.7 A note on optimal communication complexity

The string length of any computationally secure OT protocol can be polynomially extended by standard techniques (cf. protocol $\Pi_{\text{OT}}^{\text{enlarge}}$ in Figure 7.7). It is straightforward to show UC-security of this approach. Hence, optimal communication complexity of the computational versions of our OT solution is not a noteworthy result. However, applying an analogous transformation to commitments or OTMs would destroy UC-security (see Remark 44 below) and we are not aware of any universally composable amortization techniques for these primitives that do not come along with additional setup assumptions.

Remark 44. The protocols $\Pi_{\text{COM}}^{\text{enlarge}}$ and $\Pi_{\text{OTM}}^{\text{enlarge}}$ in Figure 7.7 are not UC-secure.

Proof. We just show that $\Pi_{\text{COM}}^{\text{enlarge}}$ is not UC-secure. For $\Pi_{\text{OTM}}^{\text{enlarge}}$ one can argue analogously. Consider a passively corrupted receiver party that just hands over every message to the environment. For the real model, this means that in the commit phase the environment learns some k-bit string r and in the unveil phase it learns a seed $s \in \{0,1\}^l$, such that $r \oplus F(s)$ is the honest sender party's input c. Now, if the environment chooses the honest sender party's input $c \in \{0,1\}^k$ uniformly at random, this is not simulatable in the ideal model. The simulator has to choose r before he learns c. Thus, using a simple counting argument, the probability that there exists any seed $s \in \{0,1\}^l$ with $r \oplus F(s) = c$ can be upper bounded by 2^{l-k}. In other words, the simulation fails at least with probability $1 - 2^{l-k}$. □

Protocol $\Pi_{\mathrm{OT}}^{\mathrm{enlarge}}$

Parametrized by two security parameters k and l with $k > l$, a hybrid functionality $\mathcal{F}_{\mathrm{OT}}$ for l-bit string-OT and a PRNG function F with seed length l and output length k, i.e. $F : \{0,1\}^l \to \{0,1\}^k$.

1. Upon input (s_0, s_1) from the environment, the sender party verifies that $s_0, s_1 \in \{0,1\}^k$; else that input is ignored. Next, the sender party chooses two random seeds $\tilde{s}_0, \tilde{s}_1 \xleftarrow{\mathrm{r}} \{0,1\}^l$ and inputs $(\tilde{s}_0, \tilde{s}_1)$ into $\mathcal{F}_{\mathrm{OT}}$.
2. Upon input x from the environment, the receiver party verifies that $x \in \{0,1\}$; else that input is ignored. Next, the receiver party inputs x into $\mathcal{F}_{\mathrm{OT}}$, thus receiving $\tilde{s}_x$.
3. The sender party, after being notified that everybody did provide some input to $\mathcal{F}_{\mathrm{OT}}$, announces $r_0 := s_0 \oplus F(\tilde{s}_0)$ and $r_1 := s_1 \oplus F(\tilde{s}_1)$.
4. The receiver party computes and outputs $s_x = r_x \oplus F(\tilde{s}_x)$.

Protocol $\Pi_{\mathrm{COM}}^{\mathrm{enlarge}}$

Parametrized by two security parameters k and l with $k > l$, a hybrid functionality $\mathcal{F}_{\mathrm{COM}}$ for l-bit string-commitment and a PRNG function F with seed length l and output length k, i.e. $F : \{0,1\}^l \to \{0,1\}^k$.

Commit phase:

1. Upon input $(\texttt{Commit}, c)$ from the environment, the sender party verifies that $c \in \{0,1\}^k$; else that input is ignored. Next, the sender party chooses some random $\tilde{s} \xleftarrow{\mathrm{r}} \{0,1\}^l$, commits to s via $\mathcal{F}_{\mathrm{COM}}$ and sends $r := c \oplus F(\tilde{s})$ to the receiver party.
2. The receiver party outputs (`committed`).

Unveil phase:

3. Upon input (`Unveil`) from the environment, the sender party unveils $\tilde{s}$.
4. If the unveil is successful, the receiver party computes and outputs $r \oplus F(\tilde{s})$; otherwise it outputs $\perp$.

Protocol $\Pi_{\mathrm{OTM}}^{\mathrm{enlarge}}$

Parametrized by two security parameters k and l with $k > l$, a hybrid functionality $\mathcal{F}_{\mathrm{OTM}}$ for l-bit OTM and a PRNG function F with seed length l and output length k, i.e. $F : \{0,1\}^l \to \{0,1\}^k$.

Creation:

1. Upon input (s_0, s_1) from the environment, the sender party verifies that $s_0, s_1 \in \{0,1\}^k$; else that input is ignored. Next, the sender party chooses two random seeds $\tilde{s}_0, \tilde{s}_1 \xleftarrow{\mathrm{r}} \{0,1\}^l$, sends $(\tilde{s}_0, \tilde{s}_1)$ via $\mathcal{F}_{\mathrm{OTM}}$ to the receiver party and announces $r_0 := s_0 \oplus F(\tilde{s}_0)$ and $r_1 := s_1 \oplus F(\tilde{s}_1)$.
2. The receiver party outputs (`ready`).

Query:

3. Upon input x from the environment, the receiver party verifies that $x \in \{0,1\}$; else that input is ignored. Next, the receiver party inputs x into $\mathcal{F}_{\mathrm{OTM}}$, thus receiving $\tilde{s}_x$, and computes and outputs $s_x = r_x \oplus F(\tilde{s}_x)$.

Figure 7.7: Straightforward approaches for enlarging the string length of some given OT, commitment or OTM functionality, using a PRNG. The protocol $\Pi_{\mathrm{OT}}^{\mathrm{enlarge}}$ is UC-secure, but $\Pi_{\mathrm{COM}}^{\mathrm{enlarge}}$ and $\Pi_{\mathrm{OTM}}^{\mathrm{enlarge}}$ are not (q.v. Remark 44).

8 Correctness and security of our protocol

In this section we show that in the $\mathcal{F}_{\text{wrap}}^{\text{stateful}}$-hybrid model our protocol $\Pi_{\text{OAFE}}^{\text{semi-int}}$ (q.v. Figure 7.1) is a universally composable implementation of the ideal functionality $\mathcal{F}_{\text{OAFE}}^{\text{seq-ot}}$ (q.v. Figure 6.3), if only $k \geq 5$. In particular, in Section 8.2 we will prove perfect security against a corrupted David for all k and in Section 8.3 we will prove statistical security against a corrupted Goliath for the case that $k \geq 5$. However, first of all we will show that $\Pi_{\text{OAFE}}^{\text{semi-int}}$ always works correctly when no party is corrupted (Section 8.1).

8.1 Correctness

In a totally uncorrupted setting, simulation is straightforward. Since the simulator always is notified when the ideal Goliath receives input from the environment and the simulator also may arbitrarily delay the ideal David's corresponding ready-message, he can perfectly simulate any scheduling of the messages in the send phase. In turn, the choice phase cannot be influenced by the real model adversary and therefore can be simulated trivially. Furthermore, whenever in the real model the receiver David outputs some (y_i, i), it holds that $y_i = a_i x_i + b_i$, as one can verify as follows:

$$y_i \quad = \quad G\underbrace{(r_i z_i + S_i)}_{=W_i} h_i + \underbrace{(a_i - Gr_i)}_{=\tilde{a}_i} x_i + \underbrace{b_i - GS_i h_i}_{=\tilde{b}_i} \quad = \quad Gr_i \underbrace{(z_i h_i - x_i)}_{=0} + a_i x_i + b_i$$

Also note that in a totally uncorrupted setting David's consistency checks are always passed.

8.2 Security against a corrupted receiver

We first show security against a corrupted receiver party David, as this is the easy case. Basically, there are only two things a corrupted David can do: follow the protocol honestly, or query the token before the respective send phase is over. We will refer to the former as the *regular case* and to the latter as the *irregular case*. A bit more formally, regarding some specific $i \in \{1, \ldots, n\}$ we speak of the regular case if David sends (h_i, i) to Goliath before inputting (z_i, i) into the token, and we speak of the irregular case if David sends (h_i, i) to Goliath after inputting (z_i, i) into the token. Note that, although David is corrupted, h_i and z_i are still well-defined since Goliath and the token accept only well-formed messages (h_i, i) and (z_i, i) respectively. It is quite straightforward to see that due to the randomness of (r_i, S_i) every value seen by David, namely $\tilde{r}_i, \tilde{S}_i, \tilde{a}_i, \tilde{b}_i, W_i$, is just uniformly random subject to the sole condition that in the end the correct result y_i can be computed. We formalize this by the next lemma.

Lemma 45. *In our protocol $\Pi_{\text{OAFE}}^{\text{semi-int}}$, if Goliath is honest, the variables $\tilde{r}_i, \tilde{S}_i, \tilde{a}_i, \tilde{b}_i, W_i$ are just uniformly random subject to the condition that $CW_i = \tilde{r}_i z_i + \tilde{S}_i$ and $GW_i h_i + \tilde{a}_i z_i h_i + \tilde{b}_i = a_i z_i h_i + b_i$.*

Proof. We give a proof by cases and start off with the regular case, i.e. David first sends (h_i, i) to Goliath and later on inputs (z_i, i) into the token. In this case, Goliath obviously just announces some $\tilde{r}_i, \tilde{S}_i, \tilde{a}_i, \tilde{b}_i$ uniformly at random. Consequently, we have to show now that the token output W_i is uniformly random subject to the condition that $CW_i = \tilde{r}_i z_i + \tilde{S}_i$ and $GW_i h_i + \tilde{a}_i z_i h_i + \tilde{b}_i = a_i z_i h_i + b_i$. However, we can imagine that right before the computation of W_i the token's stored randomness S_i is uniformly resampled subject to the condition that $CS_i = \tilde{S}_i$ and $GS_i h_i = b_i - \tilde{b}_i$. This clearly

does not change David's view at all. In other words, we can replace S_i by $S_i + S'$, where S' is uniformly random subject to the condition that $CS' = 0$ and $GS'h = 0$. Thereby, W_i is also replaced by $W_i + S'$ and hence becomes uniformly random subject to the sole condition that CW_i and GW_ih_i are not changed. This means that W_i is uniformly random subject to the condition that $CW_i = \tilde{r}_i z_i + \tilde{S}_i$ and $GW_ih_i = (a_i - \tilde{a}_i)z_ih_i + b_i - \tilde{b}_i$. This concludes our proof for the regular case.

Now we consider the irregular case, i.e. the corrupted David inputs (z_i, i) into the token before he sends (h_i, i) to Goliath. In this case, $(\tilde{r}_i, \tilde{S}_i)$ announced by Goliath and the token's output W_i are just uniformly random subject to the condition that $CW_i = \tilde{r}_i z_i + \tilde{S}_i$. Consequently, we have to show now that the honest Goliath's announcement of $(\tilde{a}_i, \tilde{b}_i)$ is uniformly random subject to the condition that $GW_ih_i + \tilde{a}_i z_ih_i + \tilde{b}_i = a_i z_ih_i + b_i$. However, we can imagine that right before the computation of $(\tilde{a}_i, \tilde{b}_i)$ the stored randomness r_i in Goliath's memory is uniformly resampled subject to the condition that $Cr_i = \tilde{r}_i$, and S_i is replaced by the new value of $W_i - r_i z_i$. It is straightforward to verify that this does not change the corrupted David's view at all. In other words, we can replace (r_i, S_i) by $(r_i + r', S_i - r'z_i)$, where r' is uniformly random subject to the condition that $Cr' = 0$. Thereby, $\tilde{a}_i$ is replaced by $\tilde{a}_i - Gr'$ and hence becomes just uniformly random over $\mathbb{F}_q^k$, since G is complementary to C and thus Gr' itself is uniformly random over $\mathbb{F}_q^k$. Analogously, $\tilde{b}_i$ is replaced by $\tilde{b}_i + Gr'z_ih_i$, and it still holds:

$$GW_ih_i + \underbrace{(\tilde{a}_i - Gr')}_{\text{new } \tilde{a}_i} z_ih_i + \underbrace{(\tilde{b}_i + Gr'z_ih_i)}_{\text{new } \tilde{b}_i} \;=\; GW_ih_i + \tilde{a}_i z_ih_i + \tilde{b}_i \;=\; a_i z_ih_i + b_i$$

This means that $\tilde{a}_i, \tilde{b}_i$ are uniformly random subject to the sole condition that $GW_ih_i + \tilde{a}_i z_ih_i + \tilde{b}_i = a_i z_ih_i + b_i$. This concludes our proof for the irregular case. □

This lemma leads to a very straightforward simulator construction for the UC framework. When the corrupted David queries the token *after* he already got the derandomization information $(\tilde{a}_i, \tilde{b}_i)$ from Goliath in the corresponding send phase, the simulator can revise the token's output W_i so that the check $CW_i \stackrel{?}{=} \tilde{r}_i z_i + \tilde{S}_i$ is still passed, but GW_i now matches a protocol run in the real model: When the token is to output W_i, the simulator has already seen both shares z_i, h_i that are needed to extract David's input x_i. The simulator can then query the ideal functionality $\mathcal{F}_{\text{OAFE}}^{\text{seq}-\text{ot}}$ on this input x_i, thus receiving y_i, and then revise W_i by some W' so that $CW' = CW_i$ and $y_i = GW'h_i + \tilde{a}_i x_i + \tilde{b}_i$. Note that existence of such a W' is always guaranteed, since G is complementary to C (i.e. especially G has full rank) and $h \neq 0$.

When the corrupted David queries the token *before* he got the derandomization information $(\tilde{a}_i, \tilde{b}_i)$ from Goliath in the corresponding send phase, the simulator can easily revise Goliath's announcement of the derandomization information so that it matches a protocol run in the real model: When Goliath is to announce the derandomization information $(\tilde{a}_i, \tilde{b}_i)$, the simulator has already seen both shares z_i, h_i that are needed to extract David's input x_i. The simulator can then query the ideal functionality $\mathcal{F}_{\text{OAFE}}^{\text{seq}-\text{ot}}$ on this input x_i, thus receiving y_i, and then just revise $\tilde{b}_i$ so that $y_i = GW_ih_i + \tilde{a}_i x_i + \tilde{b}_i$.

A formal description of this simulator construction is given in Figure 8.1. We conclude this section with the corresponding security theorem.

Theorem 46. *Let some arbitrary environment $\mathcal{Z}$ be given and some adversary $\mathcal{A}$ that corrupts the receiver David. Then the view of $\mathcal{Z}$ in the ideal model with ideal functionality $\mathcal{F}_{\text{OAFE}}^{\text{seq}-\text{ot}}$ and simulator $\mathcal{S}^{\text{David}}(\mathcal{A})$ is identically distributed to the view of $\mathcal{Z}$ in the real model with protocol $\Pi_{\text{OAFE}}^{\text{semi}-\text{int}}$ and adversary $\mathcal{A}$.*

Proof. This directly follows by Lemma 45. □

Simulator $\mathcal{S}^{\mathrm{David}}(\mathcal{A})$

- Set up an honest Goliath-machine $\mathcal{G}$; also set up simulated versions of $\mathcal{F}_{\mathrm{wrap}}^{\mathrm{stateful}}$ and the given real model adversary $\mathcal{A}$ (which especially impersonates the corrupted David). Wire the simulated machines $\mathcal{A}, \mathcal{G}, \mathcal{F}_{\mathrm{wrap}}^{\mathrm{stateful}}$ to each other and $\mathcal{A}$ to the environment right the way they would be wired in the real model with protocol $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$ (q.v. Figure 7.1).
- Upon receiving a message $(\texttt{created}, i)$ from the ideal functionality $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$, reply with $(\texttt{Delivery}, i)$. Then, upon receiving $(\texttt{ready}, i)$ on behalf of the corrupted David, choose some random vectors $a_i, b_i \xleftarrow{\mathrm{r}} \mathbb{F}_q^k$ and let $\mathcal{G}$ start the i-th send phase with input (a_i, b_i, i).
- Whenever $\mathcal{G}$ is to send some $(\tilde{a}_i, \tilde{b}_i, i)$ to the corrupted David in step 3 of $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$, extract the current state j of j'_{queried} from the view of the simulated $\mathcal{F}_{\mathrm{wrap}}^{\mathrm{stateful}}$. If $j \geq i$, replace the announcement $(\tilde{a}_i, \tilde{b}_i, i)$ by $(\tilde{a}_i, \tilde{b}', i)$, with $\tilde{b}'$ computed as follows:
  0. Extract G, h_i from the view of $\mathcal{G}$ and z_i, W_i from the view of the simulated $\mathcal{F}_{\mathrm{wrap}}^{\mathrm{stateful}}$.
  1. Compute $x_i \leftarrow z_i h_i$ and (on behalf of the corrupted David) send (x_i, i) to the ideal functionality $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$; let (y_i, i) denote the respective answer from $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$.
  2. Set $\tilde{b}' \leftarrow y_i - GW_i h_i - \tilde{a}_i x_i$.
- Whenever the token is to output some matrix W to the corrupted David in step 6 of $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$, extract the current state i of j'_{queried} from the view of the simulated $\mathcal{F}_{\mathrm{wrap}}^{\mathrm{stateful}}$. When $\mathcal{G}$ has already received (and not ignored) a message (h_i, i) in step 3 of $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$, replace the token's output by W', computed as follows:
  0. Extract $C, G, \tilde{r}_i, \tilde{S}_i, h_i, \tilde{a}_i, \tilde{b}_i$ from the view of $\mathcal{G}$ and z_i from the view of the simulated $\mathcal{F}_{\mathrm{wrap}}^{\mathrm{stateful}}$.
  1. Compute $x_i \leftarrow z_i h_i$ and (on behalf of the corrupted David) send (x_i, i) to the ideal functionality $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$; let (y_i, i) denote the respective answer from $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$.
  2. Choose randomly $W' \xleftarrow{\mathrm{r}} \left\{ \widetilde{W} \in \mathbb{F}_q^{4k \times k} \;\middle|\; C\widetilde{W} = \tilde{r}_i z_i + \tilde{S}_i \;\wedge\; G\widetilde{W} h_i + \tilde{a}_i x_i + \tilde{b}_i = y_i \right\}$.

Figure 8.1: The simulator program $\mathcal{S}^{\mathrm{David}}(\mathcal{A})$, given an adversary $\mathcal{A}$ that corrupts David.

8.3 Security against a corrupted sender

The case of a corrupted sender Goliath is the technically challenging part of the security proof. However, before we give our simulator construction for a corrupted sender Goliath (q.v. Section 8.3.5), we first take a closer look at the problems we have to deal with, and introduce the respective solution tools (Section 8.3.1, Section 8.3.3, Section 8.3.2 and Section 8.3.4).

8.3.1 Independence of the token view

We start our security considerations with showing that an honest David's token inputs $z_1, \ldots, z_n$ are statistically indistinguishable from uniform randomness. This is necessary for security, since otherwise David's OAFE inputs $x_1, \ldots, x_n$ would be non-negligibly correlated with the token view and a malicious token's behavior in the n-th choice phase could depend on $x_1, \ldots, x_{n-1}$.

W.l.o.g., we can assume that David's random tape is chosen after all other random tapes, i.e. we can consider everything to be deterministic except for David's random choice of $h_1, \ldots, h_n$ and $z_1, \ldots, z_n$. However, as we are aiming for universal composability, we must take into account that David's i-th OAFE input x_i might depend on everything that Goliath learned so far. In particular, all n send phases might already be over, i.e. Goliath already knows $h_1, \ldots, h_n$, and there might have leaked some little information about $z_1, \ldots, z_{i-1}$ during past choice phases. Therefore, we have to model David's i-th OAFE input x_i as a function value $x_i(h_1, \ldots, h_n, z_1, \ldots, z_{i-1})$.

Lemma 47. *Let $\mathbb{F}_q$ be some arbitrary field of size $q \geq 2$ and let $k, n \in \mathbb{N}_{>0}$. Let $\mathcal{U} := \mathbb{F}_q^{1\times k}$ and $\mathcal{H} := \mathbb{F}_q^k \setminus \{0\}$. Further, for $i = 1, \ldots, n$ let any mapping $x_i : \mathcal{H}^n \times \mathcal{U}^{i-1} \to \mathbb{F}_q$ be given. Finally, for $i = 1, \ldots, n$ we define the following random variables:*

$$\mathbf{h}_i \xleftarrow{\mathrm{r}} \mathcal{H} \qquad \mathbf{z}_i \xleftarrow{\mathrm{r}} \{z \in \mathcal{U} \mid z\mathbf{h}_i = x_i(\mathbf{h}_1, \ldots, \mathbf{h}_n, \mathbf{z}_1, \ldots, \mathbf{z}_{i-1})\} \qquad \mathbf{u}_i \xleftarrow{\mathrm{r}} \mathcal{U}$$

Then it holds that $\mathsf{SD}((\mathbf{z}_1, \ldots, \mathbf{z}_n), (\mathbf{u}_1, \ldots, \mathbf{u}_n)) < \frac{1}{2}\sqrt{\exp(n \cdot q^{2-k}) - 1}$.

Proof. We show this by estimation techniques borrowed from a proof for the Leftover Hash Lemma [AB09, proof of Lemma 21.26]. Let $\vec{z} \in \mathbb{R}^{\mathcal{U}^n}$ denote the probability vector of $(\mathbf{z}_1, \ldots, \mathbf{z}_n)$ and let $\vec{u} \in \mathbb{R}^{\mathcal{U}^n}$ denote the probability vector of $(\mathbf{u}_1, \ldots, \mathbf{u}_n)$. Note that $\vec{z} - \vec{u}$ is orthogonal to $\vec{u}$:

$$\langle \vec{z} - \vec{u} \,|\, \vec{u} \rangle \;=\; \langle \vec{z} \,|\, \vec{u} \rangle - \langle \vec{u} \,|\, \vec{u} \rangle \;=\; \tfrac{\|\vec{z}\|_1}{|\mathcal{U}^n|} - \tfrac{\|\vec{u}\|_1}{|\mathcal{U}^n|} \;=\; \tfrac{1}{|\mathcal{U}^n|} - \tfrac{1}{|\mathcal{U}^n|} \;=\; 0$$

Let the $2n$-tuple of random variables $(\mathbf{h}'_1, \ldots, \mathbf{h}'_n, \mathbf{z}'_1, \ldots, \mathbf{z}'_n)$ be identically distributed as its unprimed counterpart $(\mathbf{h}_1, \ldots, \mathbf{h}_n, \mathbf{z}_1, \ldots, \mathbf{z}_n)$. The following equation system has exactly q^{k-2} different solutions $z \in \mathcal{U}$ if $\mathbf{h}_i$ and $\mathbf{h}'_i$ are linearly independent, and at most q^{k-1} solutions otherwise:

$$\begin{aligned} z\mathbf{h}_i &= x_i(\mathbf{h}_1, \ldots, \mathbf{h}_n, \mathbf{z}_1, \ldots, \mathbf{z}_{i-1}) \\ z\mathbf{h}'_i &= x_i(\mathbf{h}'_1, \ldots, \mathbf{h}'_n, \mathbf{z}'_1, \ldots, \mathbf{z}'_{i-1}) \end{aligned}$$

Using the auxiliary random variable $\mathbf{m} := \#\{i \in \{1, \ldots, n\} \mid \mathbf{h}_i \text{ and } \mathbf{h}'_i \text{ are linearly independent}\}$, we can thus estimate:

$$\mathbb{P}\left[(\mathbf{z}_1, \ldots, \mathbf{z}_n) = (\mathbf{z}'_1, \ldots, \mathbf{z}'_n) \mid \mathbf{m} = m\right] \;\leq\; \left(\frac{q^{k-2}}{q^{k-1} \cdot q^{k-1}}\right)^m \cdot \left(\frac{q^{k-1}}{q^{k-1} \cdot q^{k-1}}\right)^{n-m}$$

Further, we have that $\mathbb{P}[\mathbf{m} = m] = \binom{n}{m} \cdot \left(\frac{|\mathcal{H}|-(q-1)}{|\mathcal{H}|}\right)^m \cdot \left(\frac{q-1}{|\mathcal{H}|}\right)^{n-m}$ by construction. It follows:

$$\begin{aligned} \|\vec{z}\|_2^2 \;&=\; \mathbb{P}\left[(\mathbf{z}_1, \ldots, \mathbf{z}_n) = (\mathbf{z}'_1, \ldots, \mathbf{z}'_n)\right] \\ &=\; \sum_{m=0}^{n} \mathbb{P}[\mathbf{m} = m] \cdot \mathbb{P}\left[(\mathbf{z}_1, \ldots, \mathbf{z}_n) = (\mathbf{z}'_1, \ldots, \mathbf{z}'_n) \mid \mathbf{m} = m\right] \\ &\leq\; \sum_{m=0}^{n} \binom{n}{m} \cdot \left(\frac{|\mathcal{H}| - (q-1)}{|\mathcal{H}|}\right)^m \cdot \left(\frac{q-1}{|\mathcal{H}|}\right)^{n-m} \cdot \left(\frac{q^{k-2}}{q^{k-1} \cdot q^{k-1}}\right)^m \cdot \left(\frac{q^{k-1}}{q^{k-1} \cdot q^{k-1}}\right)^{n-m} \\ &=\; \left(1 + \tfrac{(q-1)^2}{|\mathcal{H}|}\right)^n \cdot q^{-nk} \end{aligned}$$

Using the Pythagorean Theorem, we can now estimate:

$$\|\vec{z} - \vec{u}\|_2^2 \;=\; \|\vec{z}\|_2^2 - \|\vec{u}\|_2^2 \;\leq\; \left(1 + \tfrac{(q-1)^2}{|\mathcal{H}|}\right)^n \cdot q^{-nk} - q^{-nk}$$

Since $\|\vec{v}\|_1 \leq \sqrt{m} \cdot \|\vec{v}\|_2$ for all $m \in \mathbb{N}$, $\vec{v} \in \mathbb{R}^m$, this yields:

$$\mathsf{SD}((\mathbf{z}_1, \ldots, \mathbf{z}_n), (\mathbf{u}_1, \ldots, \mathbf{u}_n)) \;=\; \tfrac{1}{2}\|\vec{z} - \vec{u}\|_1 \;\leq\; \tfrac{1}{2}\sqrt{|\mathcal{U}^n|} \cdot \|\vec{z} - \vec{u}\|_2 \;\leq\; \tfrac{1}{2}\sqrt{\left(1 + \tfrac{(q-1)^2}{|\mathcal{H}|}\right)^n - 1}$$

To conclude our proof, we further estimate:

$$\left(1 + \tfrac{(q-1)^2}{|\mathcal{H}|}\right)^n \;=\; \exp\left(n \cdot \ln\left(1 + \tfrac{(q-1)^2}{|\mathcal{H}|}\right)\right) \;<\; \exp\left(\tfrac{n\cdot(q-1)^2}{|\mathcal{H}|}\right) \;<\; \exp\left(n \cdot q^{2-k}\right) \qquad \square$$

We will use this lemma not directly but for showing that the token functionality in the $(m+1)$-th choice phase can be considered to be independent of C and $h_{m+1}, \ldots, h_n$. So in the following corollary, the random variable $\mathbf{R}$ can be thought of as David's random choice of $(C, h_{m+1}, \ldots, h_n)$.

Corollary 48. *Let $\mathbb{F}_q$ be some arbitrary field of size $q \geq 2$ and let $k, m \in \mathbb{N}_{>0}$. Let $\mathcal{U} := \mathbb{F}_q^{1\times k}$ and $\mathcal{H} := \mathbb{F}_q^k \setminus \{0\}$. Further, let $\mathbf{R}$ be some arbitrary random variable with finite support $\mathcal{R}$. For $i = 1, \ldots, m$ let any mapping $x_i : \mathcal{R} \times \mathcal{H}^m \times \mathcal{U}^{i-1} \to \mathbb{F}_q$ be given. Finally, for $i = 1, \ldots, m$ we define the following random variables:*

$$\mathbf{h}_i \xleftarrow{\text{r}} \mathcal{H} \qquad\qquad \mathbf{z}_i \xleftarrow{\text{r}} \{z \in \mathcal{U} \mid z\mathbf{h}_i = x_i(\mathbf{R}, \mathbf{h}_1, \ldots, \mathbf{h}_m, \mathbf{z}_1, \ldots, \mathbf{z}_{i-1})\}$$

Then it holds that $\iota(\mathbf{R}, (\mathbf{z}_1, \ldots, \mathbf{z}_m)) < \sqrt{\exp(mq^{2-k}) - 1}$.

Proof. By the Triangle Inequality (and the definition of ι, q.v. Section 6.1), we already have that $\iota(\mathbf{R}, (\mathbf{z}_1, \ldots, \mathbf{z}_m)) \leq 2 \cdot \mathsf{SD}((\mathbf{z}_1, \ldots, \mathbf{z}_m), (\mathbf{u}_1, \ldots, \mathbf{u}_m))$ for any random variables $\mathbf{u}_1, \ldots, \mathbf{u}_m$ that are statistically independent from $\mathbf{R}$. Hence, our corollary directly follows by Lemma 47. □

After all, note that independence between the token view and David's OAFE inputs $x_1, \ldots, x_n$ is only a starting point for our security proof. E.g., we have not used so far David's consistency check $CW_i \stackrel{?}{=} \tilde{r}_i z_i + \tilde{S}_i$ in the final step of the choice phases. Lemma 47 and Corollary 48 would still hold true without this consistency check, but the protocol would become susceptible to attacks where the token encodes z_{i-1} into W_i. Now, if this information about z_{i-1} is unveiled to Goliath, e.g. through the unveil message in a commitment protocol (q.v. Protocol $\Pi_{\text{COM}}^{\text{backward}}$ in Figure 7.6), he can possibly reconstruct David's secret OAFE input x_{i-1}, although the token still learns nothing but uniform randomness. Thus, what we have shown so far can only be one core argument amongst several others.

8.3.2 Committing the token to affine behavior

In Section 7.1 we argued that David's check $CW_i \stackrel{?}{=} \tilde{r}_i z_i + \tilde{S}_i$ enforces affine behavior of the token, since otherwise the token could form collisions for the universal hash function C. However, this is only half the truth. In fact, with $\tau_i : \mathbb{F}_q^{1\times k} \to \mathbb{F}_q^{4k\times k}$ denoting the token functionality in the i-th choice phase, for each possible token input $z \in \mathbb{F}_q^{1\times k}$ there are always exactly q^{4k} different parameter tuples $(r, S) \in \mathbb{F}_q^{4k} \times \mathbb{F}_q^{4k\times k}$, such that $\tau_i(z) = rz + S$. In particular, for all $z \in \mathbb{F}_q^{1\times k}$, $r \in \mathbb{F}_q^{4k}$ we can complement r to a matching parameter tuple by $S := \tau_i(z) - rz$. In total, there might exist up to q^{5k} different parameter tuples belonging to any image of τ_i and we must somehow rule out that there are too many collisions of the form $(Cr, CS) = (Cr', CS')$ with distinct $(r, S), (r', S')$.

Since the space of potential parameter tuples is that large, pure counting arguments (e.g. by considering the random matrix C as a 2-universal hash function) cannot be sufficient as long as the special structure of our problem is ignored: For example, consider the hypothetical case that every parameter tuple (r, S) has to be taken into account where each column of S equals r. Although this yields only q^{4k} different parameter tuples, we would always have that equivalence classes of q^k different parameter tuples do collide. However, this is exactly the size of the preimage space of τ_i. Thereby we just cannot rule out that τ_i is non-affine on every $Z \subseteq \mathbb{F}_q^{1\times k}$ with $|Z| > 1$, but enough parameter tuples collide so that $C \cdot \tau_i$ is affine on the complete input space $\mathbb{F}_q^{1\times k}$. Note also that this problem cannot be circumvent by enlarging the token input space to $\mathbb{F}_q^{1\times \alpha k}$ for some $\alpha > 1$, since in that case we can still argue analogously with the condition $|Z| > 1$ replace by $|Z| > q^{(\alpha-1)k}$.

So, we explicitly have to exploit that the space of affine mappings $\mathbb{F}_q^{1\times k} \to \mathbb{F}_q^{4k\times k}$ has some specific structure. In fact, we only need the random matrix C to have some rank-preserving property when operating on the image space of τ_i. Given this (and a not too large overall abortion probability in the current choice phase), we can show that τ_i is affine on *all* token inputs that do not cause a protocol abortion.

Lemma 49. *Let $\mathbb{F}_q$ be some finite field of size $q \geq 2$ and let $l, m, k \in \mathbb{N}_{>0}$. Let $\tau : \mathbb{F}_q^{1\times k} \to \mathbb{F}_q^{m\times k}$ be some arbitrary mapping and let $C \in \mathbb{F}_q^{l\times m}$, $\tilde{r} \in \mathbb{F}_q^{l}$, $\tilde{S} \in \mathbb{F}_q^{l\times k}$, $V := \{v \in \mathbb{F}_q^{1\times k} \mid C\cdot\tau(v) = \tilde{r}\cdot v + \tilde{S}\}$, such that $|V| > q$ and for all $v, v' \in V$ the following implications hold true:*

$$\begin{aligned} \operatorname{rank}(\tau(v) - \tau(v')) > 0 \quad &\Rightarrow \quad \operatorname{rank}(C\cdot\tau(v) - C\cdot\tau(v')) > 0 \\ \operatorname{rank}(\tau(v) - \tau(v')) > 1 \quad &\Rightarrow \quad \operatorname{rank}(C\cdot\tau(v) - C\cdot\tau(v')) > 1 \end{aligned}$$

Then there exists a unique tuple $(r, S) \in \mathbb{F}_q^m \times \mathbb{F}_q^{m\times k}$, such that $\tau(v) = r\cdot v + S$ for all $v \in V$. Further, for this unique tuple it holds that $(Cr, CS) = (\tilde{r}, \tilde{S})$.

Proof. We just need to show existence of (r, S); everything else follows straightforwardly. Moreover, if $\tilde{r} = 0$, the proof is trivial. In this case, since by assumption $\tau(v) = \tau(v')$ for all $v, v' \in V$ with $C\cdot\tau(v) = C\cdot\tau(v')$, we have that τ is constant on the entire input set V. So, w.l.o.g. let $\tilde{r} \neq 0$.

First of all, we now observe for all $v, v' \in V$ that $\operatorname{rank}(\tau(v) - \tau(v')) \leq 1$, since else by the rank-preserving properties of C we had the contradiction that $1 < \operatorname{rank}(C\cdot\tau(v) - C\cdot\tau(v')) = \operatorname{rank}(\tilde{r}\cdot(v - v')) \leq 1$. Thereby, for all $v, v' \in V$ we find some $r \in \mathbb{F}_q^m$, $\bar{v} \in \mathbb{F}_q^{1\times n}$, such that $\tau(v) - \tau(v') = r\cdot\bar{v}$. Moreover, we can always choose $\bar{v} := v - v'$, since $\tilde{r}\cdot(v - v') = Cr\cdot\bar{v}$ and we assumed that $\tilde{r} \neq 0$. Thus we have:

$$\forall\, v, v' \in V\ \exists\, r \in \mathbb{F}_q^m :\ \tau(v) - \tau(v') = r\cdot(v - v')$$

We will show now that r in fact is independent of v, v'. More precisely, we will show that for arbitrary $v, v', v'' \in V$ with linearly independent $v - v', v' - v''$ there always exists an $r \in \mathbb{F}_q^m$, such that $\tau(v) - \tau(v') = r\cdot(v - v')$ and $\tau(v') - \tau(v'') = r\cdot(v' - v'')$. It is sufficient to consider the case of linearly independent $v - v', v' - v''$, since $|V| > q$ by assumption and hence the affine span of V must have dimension 2 or higher; therefore for all $v, v', v'' \in V$ with linearly dependent $v - v', v' - v''$ there exists some $\hat{v} \in V$, such that $v - v', v' - \hat{v}$ are linearly independent and also are $\hat{v} - v', v' - v''$. So, let any $v, v', v'' \in V$, $r, r' \in \mathbb{F}_q^m$ be given with linearly independent $v - v', v' - v''$ and:

$$\begin{aligned} \tau(v) - \tau(v') &= r\cdot(v - v') \\ \tau(v') - \tau(v'') &= r'\cdot(v' - v'') \end{aligned}$$

Thereby follows:

$$\operatorname{rank}(r\cdot(v - v') + r'\cdot(v' - v'')) \quad = \quad \operatorname{rank}(\tau(v) - \tau(v'')) \quad \leq \quad 1$$

Since $v - v', v' - v''$ are linearly independent, this yields that r, r' must be linearly dependent. Hence, on the one hand we find some $\hat{r} \in \mathbb{F}_q^m$ and $\alpha, \alpha' \in \mathbb{F}_q$, such that $r = \alpha\hat{r}$ and $r' = \alpha'\hat{r}$. On the other hand, since $v, v' \in V$, we also have:

$$\tilde{r}\cdot(v - v') \quad = \quad C\cdot\big(\tau(v) - \tau(v')\big) \quad = \quad Cr\cdot(v - v')$$

Since $v - v' \neq 0$, this yields that $Cr = \tilde{r}$ and analogously it must hold that $Cr' = \tilde{r}$. Thus we have that $\alpha C\hat{r} = \alpha' C\hat{r} = \tilde{r}$. Since we assumed that $\tilde{r} \neq 0$, we can conclude that $\alpha = \alpha'$ and hence $r = r'$.

So, once we have shown that r is unique, we can finally pick some arbitrary $\tilde{v} \in V$ and set $S := \tau(\tilde{v}) - r\cdot\tilde{v}$, whereby for every $\tilde{v}' \in M$ it follows:

$$\tau(\tilde{v}') \quad = \quad \big(\tau(\tilde{v}') - \tau(\tilde{v})\big) + \tau(\tilde{v}) \quad = \quad \big(r\cdot(\tilde{v}' - \tilde{v})\big) + \big(r\cdot\tilde{v} + S\big) \quad = \quad r\cdot\tilde{v}' + S \qquad \square$$

To make this lemma applicable, yet we need to show that with overwhelming probability the random matrix C has the required rank-preserving properties. As a formal preparation we state our next technical lemma, where the matrix set $\mathcal{W}$ should be thought of as all possible token output differences $\tau_i(z) - \tau_i(z')$ and $l := 3k$ and $m := 4k$. Thereby, since $|\mathcal{W}| < |\text{image}(\tau_i)|^2 \leq q^{2k}$, we get that the random matrix C has the required rank-preserving properties with overwhelming probability. For convenience, we state this lemma only for the case that the random matrix C is statistically independent from the token functionality τ_i (and $\mathcal{W}$ respectively). However, the error introduced by this assumption can be estimated by $\iota(C, \tau_i)$, and then Corollary 48 does apply.

Lemma 50. *Let $\mathbb{F}_q$ be some finite field of size $q \geq 2$ and let $l, m, k, r \in \mathbb{N}_{>0}$ with $r \leq \min(l, m, k)$. Then for arbitrary $\mathcal{W} \subseteq \mathbb{F}_q^{m \times k}$ and $\mathbf{C} \xleftarrow{r} \mathbb{F}_q^{l \times m}$ it holds:*

$$\mathbb{P}\left[\exists\, W \in \mathcal{W} :\ \operatorname{rank}(W) \geq r > \operatorname{rank}(\mathbf{C}W)\right] \ <\ q^{r-l}|\mathcal{W}|$$

Proof. We first estimate the number of matrices in $\mathbb{F}_q^{l \times r}$ that have full rank r. Given $i \in \{1, \ldots, r\}$ and any matrix $C \in \mathbb{F}_q^{l \times i}$ with full rank i, there exist exactly $q^l - q^i$ columns in $\mathbb{F}_q^l$ (only the linear combinations of the columns of C are excluded) by which we can extend C to a matrix of dimension $l \times (i+1)$ and rank $i+1$. By induction on i follows:

$$\#\{C \in \mathbb{F}_q^{l \times r} \mid \operatorname{rank}(C) = r\} \ =\ \prod_{i=0}^{r-1} q^l - q^i$$

Since the term $\prod_{i=0}^{r-1} q^l - q^i$ is a bit unhandy, we estimate it from below:

$$\prod_{i=0}^{r-1} q^l - q^i \ =\ q^{lr} \prod_{i=0}^{r-1} 1 - q^{i-l} \ \geq\ q^{lr}\left(1 - \sum_{i=0}^{r-1} q^{i-l}\right) \ =\ q^{lr}\left(1 - \frac{q^r - 1}{q^l(q-1)}\right) \ >\ q^{lr}\left(1 - q^{r-l}\right)$$

Now, let $W \in \mathbb{F}_q^{m \times k}$ be some arbitrary matrix with $\bar{r} := \operatorname{rank}(W) \geq r$. Further let $\bar{B} \in \mathbb{F}_q^{\bar{r} \times k}$, such that $\bar{B}$ only consists of linearly independent rows of W; i.e. especially $\bar{B}$ has full rank $\bar{r}$. Let $B \in \mathbb{F}_q^{m \times k}$, such that the first $\bar{r}$ rows of B are $\bar{B}$ and the rest of B is all-zero. Note that we can find an invertible matrix $M \in \mathbb{F}_q^{m \times m}$, such that $W = MB$. Hence we can estimate:

$$\begin{aligned}
\#\{C \in \mathbb{F}_q^{l \times m} \mid \operatorname{rank}(CW) < r\} \ &=\ q^{lm} - \#\{C \in \mathbb{F}_q^{l \times m} \mid \operatorname{rank}(CW) \geq r\} \\
&=\ q^{lm} - \#\{C \in \mathbb{F}_q^{l \times m} \mid \operatorname{rank}(CMB) \geq r\} \\
&=\ q^{lm} - \#\{C \in \mathbb{F}_q^{l \times m} \mid \operatorname{rank}(CB) \geq r\} \\
&=\ q^{lm} - \#\{C \in \mathbb{F}_q^{l \times \bar{r}} \mid \operatorname{rank}(C\bar{B}) \geq r\} \cdot q^{l(m-\bar{r})} \\
&=\ q^{lm} - \#\{C \in \mathbb{F}_q^{l \times \bar{r}} \mid \operatorname{rank}(C) \geq r\} \cdot q^{l(m-\bar{r})} \\
&\leq\ q^{lm} - \#\{C \in \mathbb{F}_q^{l \times r} \mid \operatorname{rank}(C) = r\} \cdot q^{l(m-r)} \\
&<\ q^{lm} - q^{lm}\left(1 - q^{r-l}\right) \\
&=\ q^{lm+r-l}
\end{aligned}$$

Thereby, for arbitrary $W \in \mathbb{F}_q^{m \times n}$ and $\mathbf{C} \xleftarrow{r} \mathbb{F}_q^{l \times m}$ we can conclude:

$$\mathbb{P}\left[\operatorname{rank}(W) \geq r > \operatorname{rank}(\mathbf{C}W)\right] \ <\ q^{r-l}$$

The assertion of our lemma now follows by the Union Bound. □

Basically, in each choice phase of our protocol $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$ (with honest receiver party David) we have now with overwhelming probability one of the following two cases:

- Either $\#\{z \in \mathbb{F}_q^{1 \times k} \mid C \cdot \tau_i(z) = \tilde{r}_i z + \tilde{S}_i\} \leq q$, i.e. only few token inputs pass David's consistency check and thus the protocol is aborted with overwhelming probability,
- or there exist some $r \in \mathbb{F}_q^{4k}$, $S \in \mathbb{F}_q^{4k \times k}$, such that $\tau_i(z) = rz + S$ for all $z \in \mathbb{F}_q^{1 \times k}$ with $C \cdot \tau_i(z) = \tilde{r}_i z + \tilde{S}_i$, i.e. the token functionality is affine on all inputs that pass the consistency check.

8.3.3 Uniqueness of affine approximations of the token functionality

By our technical tools developed so far, we already have that the token functionality in each choice phase is piecewise affine, and the protocol is aborted if the affine pieces are too small. However, for our formal security proof we will need that the larger affine pieces yield a disjoint decomposition of the preimage space. This motivates our next lemma.

Lemma 51. *Let $\mathbb{F}_q$ be some finite field of size $q \geq 2$, let $\varepsilon > 0$ and let $k, l \in \mathbb{N}_{>0}$, such that $q^k \geq 2^{1/\varepsilon}$. Further, let $\tau : \mathbb{F}_q^{1\times k} \to \mathbb{F}_q^{l\times k}$ be an arbitrary mapping and let V' denote the set of all $v \in \mathbb{F}_q^{1\times k}$ for that exist more than one tuple $(r, S) \in \mathbb{F}_q^l \times \mathbb{F}_q^{l\times k}$ with the following property:*

$$\tau(v) = rv + S \quad \text{and} \quad \#\{\tilde{v} \in \mathbb{F}_q^{1\times k} \mid \tau(\tilde{v}) = r\tilde{v} + S\} \geq q^{(2/3+\varepsilon)k}$$

Then we have that $|V'| < q^{2k/3}$.

Proof. We call a mapping $\gamma : \mathbb{F}_q^{1\times k} \to \mathbb{F}_q^{l\times k}$ a *straight line*, if there exist $r \in \mathbb{F}_q^k$ and $S \in \mathbb{F}_q^{l\times k}$, such that $\gamma(v) = rv + S$ for all $v \in \mathbb{F}_q^{1\times k}$. Given two straight lines γ, γ', we call $v \in \mathbb{F}_q^{1\times k}$ an *intersection* of γ and γ', if $\gamma(v) = \gamma'(v)$. Given a straight line γ, we say that γ *intersects* with τ for m times, if $\tau(v) = \gamma(v)$ for exactly m different $v \in \mathbb{F}_q^{1\times k}$. Note that two straight lines are identical, iff they have two or more common intersections.

Now, let Γ denote the set of all straight lines that intersect with τ for at least $q^{(2/3+\varepsilon)k}$ times. Thus, V' is a subset of all intersections of distinct straight lines $\gamma, \gamma' \in \Gamma$. However, as two distinct straight lines may have no more than one common intersection, m straight lines have always less than m^2 intersections in total. Thus, if $|V'| \geq q^{2k/3}$, there would be more than $q^{k/3}$ straight lines in Γ, i.e. we could find some $\Gamma' \subseteq \Gamma$ with $|\Gamma'| = \lceil q^{k/3} \rceil$. However, this leads to a contradiction, as one can see as follows. Each of the straight lines in Γ' has less than $q^{k/3}$ intersections with all the other straight lines in Γ', what leaves more than $q^{(2/3+\varepsilon)k} - q^{k/3}$ intersections with τ that are not shared with other straight lines in Γ'. Hence, overall τ must have more than $\lceil q^{k/3} \rceil \cdot (q^{(2/3+\varepsilon)k} - q^{k/3})$ of such non-shared intersections with straight lines in Γ', i.e. $|\tau(\mathbb{F}_q^{1\times k})| > q^{(1+\varepsilon)k} - q^{2k/3}$. Since $q^{\varepsilon k} \geq 2$ by assumption and thus $q^{(1+\varepsilon)k} - q^{2k/3} \geq q^k(2 - q^{-k/3}) > q^k$, this is impossible. □

8.3.4 Utilizing the Leftover Hash Lemma

We introduce now our final technical tool, a fairly technical partitioning argument, which will be needed to show that the abort behavior in the real model is indistinguishable from the abort behavior in the ideal model. Before we take a closer look at the technical details, we briefly recap the involved elements of our protocol $\Pi_{\text{OAFE}}^{\text{semi-int}}$ (q.v. Section 7.1):

- First of all, by programming the token the adversary commits to a disjoint decomposition of the i-th round token input space $\mathbb{F}_q^{1\times k}$, in the sense that each part of this decomposition corresponds to another affine token behavior in a later round.
- Next, the honest David announces a 2-universal hash function $h_i \xleftarrow{r} \mathbb{F}_q^k \setminus \{0\}$.
- Then, Goliath announces some check information corresponding to one part of the disjoint decomposition he committed to. The protocol will be aborted, iff David's i-th token input z_i is not an element of this part.
- Finally, David gets his i-th OAFE input $x_i \in \mathbb{F}_q$ from the environment, inputs a share $z_i \xleftarrow{r} \{\tilde{z} \in \mathbb{F}_q^{1\times k} \mid \tilde{z}h_i = x_i\}$ into the token, and aborts the protocol if the token output does not match Goliath's check information.

One the one hand, since the environment w.l.o.g. knows how the corrupted Goliath programmed the token and what check information he announced, the environment can exactly determine the

abort probability in the real model. On the other hand, since the simulator does not know the ideal David's OAFE input x_i, the abort probability in the ideal model is independent of x_i. Thus, we have to show that the abort probability in the real model also is not noticeably correlated with x_i. A bit more formally, the security proof basically boils down to the following problem:

- There are two adversarially chosen mappings, $x : \mathbb{F}_q^k \setminus \{0\} \to \mathbb{F}_q$ and $A : \mathbb{F}_q^k \setminus \{0\} \to \mathcal{P}(\mathbb{F}_q^{1\times k})$ (with $\mathcal{P}(\mathbb{F}_q^{1\times k})$ denoting the power set of $\mathbb{F}_q^{1\times k}$), such that for all $h, h' \in \mathbb{F}_q^k \setminus \{0\}$ the following implication holds true:

$$A(h) \neq A(h') \quad \Rightarrow \quad A(h) \cap A(h') = \emptyset$$

- There are the following three random variables:

$$\mathbf{h} \xleftarrow{\text{r}} \mathbb{F}_q^k \setminus \{0\} \qquad \mathbf{z} \xleftarrow{\text{r}} \{\tilde{z} \in \mathbb{F}_q^{1\times k} \mid \tilde{z}\mathbf{h} = x(\mathbf{h})\} \qquad \mathbf{u} \xleftarrow{\text{r}} \mathbb{F}_q^{1\times k}$$

- We have to show that $\mathsf{SD}((\mathbf{z}\in A(\mathbf{h}), \mathbf{h}), (\mathbf{u}\in A(\mathbf{h}), \mathbf{h}))$ is negligible, where $\mathbf{z}\in A(\mathbf{h})$ denotes a predicate that is true iff $\mathbf{z}$ is an element of $A(\mathbf{h})$, and analogously for $\mathbf{u}\in A(\mathbf{h})$.

We address this problem by showing that $\mathbf{h}$ partitions $A(\mathbf{h})$ into parts of roughly equal size. Our starting point is the well-known Leftover Hash Lemma, which we first recap for the sake of self-containedness (Lemma 52). If $x(\mathbf{h})$ and $A(\mathbf{h})$ were independent of $\mathbf{h}$, the Leftover Hash Lemma would already suffice to straightforwardly solve our problem. However, our problem is more complex and there seems no apparent way to directly apply the Leftover Hash Lemma. Nonetheless, we can utilize the Leftover Hash Lemma to get an estimation for the case that only $A(\mathbf{h})$ is independent of $\mathbf{h}$; q.v Lemma 53. Finally, this estimation is used to develop our technical partitioning argument (Corollary 54).

Lemma 52 (Leftover Hash Lemma [BBR88, ILL89])**.** *Let $\mathcal{G}$ be a 2-universal class of functions $\mathcal{X} \to \mathcal{Y}$ and let $\mathbf{g} \xleftarrow{\text{r}} \mathcal{G}$, i.e. for any distinct $x, x' \in \mathcal{X}$ it holds that $\mathbb{P}[\mathbf{g}(x) = \mathbf{g}(x')] \leq \frac{1}{|\mathcal{Y}|}$. Further let $\mathbf{x} \in \mathcal{X}$ be some random variable with collision entropy $\mathbb{H}_2(\mathbf{x})$. Then, if $\mathbf{x}$ and $\mathbf{g}$ are independent, for the statistical distance between $(\mathbf{g}(\mathbf{x}), \mathbf{g})$ and uniform randomness $(\mathbf{u}, \mathbf{g})$, i.e. $\mathbf{u} \xleftarrow{\text{r}} \mathcal{Y}$, it holds:*

$$\mathsf{SD}((\mathbf{g}(\mathbf{x}), \mathbf{g}), (\mathbf{u}, \mathbf{g})) \;\leq\; \tfrac{1}{2}\sqrt{2^{-\mathbb{H}_2(\mathbf{x})} \cdot |\mathcal{Y}|}$$

Proof. We adapt the proof from [AB09, proof of Lemma 21.26]. Let $(\mathbf{g}', \mathbf{x}')$ be identically distributed as its unprimed counterpart $(\mathbf{g}, \mathbf{x})$. Thereby, when we treat the distribution of $(\mathbf{g}(\mathbf{x}), \mathbf{g})$ as a probability vector $\vec{p} \in \mathbb{R}^{\mathcal{Y}\times\mathcal{G}}$, we get:

$$\begin{aligned}
\|\vec{p}\|_2^2 &= \mathbb{P}[(\mathbf{g}(\mathbf{x}), \mathbf{g}) = (\mathbf{g}'(\mathbf{x}'), \mathbf{g}')] \\
&= \mathbb{P}[\mathbf{g} = \mathbf{g}'] \cdot \mathbb{P}[\mathbf{g}(\mathbf{x}) = \mathbf{g}'(\mathbf{x}') \,|\, \mathbf{g} = \mathbf{g}'] \\
&= \mathbb{P}[\mathbf{g} = \mathbf{g}'] \cdot \mathbb{P}[\mathbf{g}(\mathbf{x}) = \mathbf{g}(\mathbf{x}')] \\
&= \mathbb{P}[\mathbf{g} = \mathbf{g}'] \cdot (\mathbb{P}[\mathbf{x} = \mathbf{x}'] + \mathbb{P}[\mathbf{x} \neq \mathbf{x}'] \cdot \mathbb{P}[\mathbf{g}(\mathbf{x}) = \mathbf{g}(\mathbf{x}') \,|\, \mathbf{x} \neq \mathbf{x}']) \\
&\leq \mathbb{P}[\mathbf{g} = \mathbf{g}'] \cdot (\mathbb{P}[\mathbf{x} = \mathbf{x}'] + \mathbb{P}[\mathbf{g}(\mathbf{x}) = \mathbf{g}(\mathbf{x}') \,|\, \mathbf{x} \neq \mathbf{x}']) \\
&= |\mathcal{G}|^{-1} \cdot \left(2^{-\mathbb{H}_2(\mathbf{x})} + \mathbb{P}[\mathbf{g}(\mathbf{x}) = \mathbf{g}(\mathbf{x}') \,|\, \mathbf{x} \neq \mathbf{x}']\right) \\
&\leq |\mathcal{G}|^{-1} \cdot \left(2^{-\mathbb{H}_2(\mathbf{x})} + |\mathcal{Y}|^{-1}\right)
\end{aligned}$$

Now, let $\vec{u} \in \mathbb{R}^{\mathcal{Y}\times\mathcal{G}}$ denote the probability vector corresponding to the uniform distribution over $\mathcal{Y}\times\mathcal{G}$. Note that $\vec{p} - \vec{u}$ is orthogonal to $\vec{u}$:

$$\langle \vec{p} - \vec{u} \,|\, \vec{u} \rangle \;=\; \langle \vec{p} \,|\, \vec{u} \rangle - \langle \vec{u} \,|\, \vec{u} \rangle \;=\; \tfrac{\|\vec{p}\|_1}{|\mathcal{Y}\times\mathcal{G}|} - \tfrac{\|\vec{u}\|_1}{|\mathcal{Y}\times\mathcal{G}|} \;=\; \tfrac{1}{|\mathcal{Y}\times\mathcal{G}|} - \tfrac{1}{|\mathcal{Y}\times\mathcal{G}|} \;=\; 0$$

By the Pythagorean Theorem follows:

$$\|\vec{p}-\vec{u}\|_2^2 \quad = \quad \|\vec{p}\|_2^2 - \|\vec{u}\|_2^2 \quad \leq \quad |\mathcal{G}|^{-1}\cdot\left(2^{-\mathbb{H}_2(\mathbf{x})}+|\mathcal{Y}|^{-1}\right)-|\mathcal{Y}\times\mathcal{G}|^{-1} \quad = \quad |\mathcal{G}|^{-1}\cdot 2^{-\mathbb{H}_2(\mathbf{x})}$$

Finally, since $\|\vec{v}\|_1 \leq \sqrt{m}\cdot\|\vec{v}\|_2$ for all $m\in\mathbb{N}$, $\vec{v}\in\mathbb{R}^m$, we can conclude:

$$\mathsf{SD}((\mathbf{g}(\mathbf{x}),\mathbf{g}),(\mathbf{u},\mathbf{g})) \;=\; \tfrac{1}{2}\|\vec{p}-\vec{u}\|_1 \;\leq\; \tfrac{1}{2}\sqrt{|\mathcal{Y}\times\mathcal{G}|}\cdot\|\vec{p}-\vec{u}\|_2 \;\leq\; \tfrac{1}{2}\sqrt{2^{-\mathbb{H}_2(\mathbf{x})}\cdot|\mathcal{Y}|} \qquad \square$$

Lemma 53. *Let $\mathbb{F}_q$ be some finite field of size $q\geq 2$ and let $k\in\mathbb{N}_{>0}$. For each $\alpha\in\mathbb{F}_q$, $h\in\mathbb{F}_q^k$ let $Z_\alpha(h) := \{z\in\mathbb{F}_q^{1\times k} \mid zh=\alpha\}$. Further, let $x:\mathcal{H}\to\mathbb{F}_q$ be some arbitrary mapping. Then, for $\mathbf{h}\xleftarrow{\text{r}}\mathcal{H}:=\mathbb{F}_q^k\setminus\{0\}$ and arbitrary $A\subseteq\mathbb{F}_q^{1\times k}$ it holds:*

$$\mathbb{E}\left|\left|A\cap Z_{x(\mathbf{h})}(\mathbf{h})\right| - \tfrac{1}{q}|A|\right| \;\leq\; \sqrt{q\cdot|A|}$$

Proof. W.l.o.g., $A\neq\emptyset$. Let $\mathbf{a}\xleftarrow{\text{r}}A$ and $\mathbf{u}\xleftarrow{\text{r}}\mathbb{F}_q$. On the one hand, since $\mathbb{P}[a\mathbf{h}=a'\mathbf{h}] = \mathbb{P}[(a-a')\mathbf{h}=0] = \frac{q^{k-1}-1}{q^k-1} < \frac{1}{|\mathbb{F}_q|}$ for all distinct $a,a'\in A$, we can estimate the statistical distance $\mathsf{SD}((\mathbf{ah},\mathbf{h}),(\mathbf{u},\mathbf{h}))$ by the Leftover Hash Lemma (Lemma 52) as follows:

$$\mathsf{SD}((\mathbf{ah},\mathbf{h}),(\mathbf{u},\mathbf{h})) \quad\leq\quad \tfrac{1}{2}\sqrt{2^{-\mathbb{H}_2(\mathbf{a})}\cdot|\mathbb{F}_q|} \quad = \quad \tfrac{1}{2}\sqrt{\tfrac{q}{|A|}}$$

On the other hand, we have:

$$\begin{aligned}
\mathsf{SD}((\mathbf{ah},\mathbf{h}),(\mathbf{u},\mathbf{h})) \;&=\; \tfrac{1}{2}\textstyle\sum_{\alpha\in\mathbb{F}_q,\,h\in\mathcal{H}}\left|\mathbb{P}[\mathbf{ah}=\alpha \;\wedge\; \mathbf{h}=h]-\mathbb{P}[\mathbf{u}=\alpha \;\wedge\; \mathbf{h}=h]\right| \\
&=\; \tfrac{1}{2}\textstyle\sum_{\alpha\in\mathbb{F}_q,\,h\in\mathcal{H}}\mathbb{P}[\mathbf{h}=h]\cdot\left|\mathbb{P}[\mathbf{a}h=\alpha]-\mathbb{P}[\mathbf{u}=\alpha]\right| \\
&=\; \tfrac{1}{2}\textstyle\sum_{\alpha\in\mathbb{F}_q,\,h\in\mathcal{H}}\mathbb{P}[\mathbf{h}=h]\cdot\left|\frac{|A\cap Z_\alpha(h)|}{|A|}-\frac{1}{q}\right| \\
&\geq\; \tfrac{1}{2}\textstyle\sum_{h\in\mathcal{H}}\mathbb{P}[\mathbf{h}=h]\cdot\left|\frac{|A\cap Z_{x(h)}(h)|}{|A|}-\frac{1}{q}\right| \\
&=\; \tfrac{1}{2}\,\mathbb{E}\left|\frac{|A\cap Z_{x(\mathbf{h})}(\mathbf{h})|}{|A|}-\frac{1}{q}\right|
\end{aligned}$$

By the linearity of expected values follows:

$$\mathbb{E}\left|\left|A\cap Z_{x(\mathbf{h})}(\mathbf{h})\right| - \tfrac{1}{q}|A|\right| \quad\leq\quad 2\,|A|\cdot\mathsf{SD}((\mathbf{ah},\mathbf{h}),(\mathbf{u},\mathbf{h})) \quad\leq\quad \sqrt{q\cdot|A|} \qquad \square$$

Corollary 54. *Let $\mathbb{F}_q$ be some finite field of size $q\geq 2$ and let $k\in\mathbb{N}_{>0}$. Let $\mathcal{H}:=\mathbb{F}_q^k\setminus\{0\}$ and let $\mathcal{R},\mathcal{Q}$ be some arbitrary finite sets. Moreover, let some mapping $A:\mathcal{R}\times\mathcal{Q}\to\mathcal{P}(\mathbb{F}_q^{1\times k})$ be given, such that for all $\nu,\nu'\in\mathcal{R}$, $t\in\mathcal{Q}$ the following implication holds true:*

$$A(\nu,t)\neq A(\nu',t) \quad\Rightarrow\quad A(\nu,t)\cap A(\nu',t)=\emptyset$$

For each $(\alpha,h)\in\mathbb{F}_q\times\mathcal{H}$ let $Z_\alpha(h) := \{z\in\mathbb{F}_q^{1\times k} \mid zh=\alpha\}$. Finally, let $\mathbf{h}\xleftarrow{\text{r}}\mathcal{H}$. Then for every random variable $\mathbf{t}\in\mathcal{Q}$ and arbitrary $\gamma\in\mathbb{R}_{>0}$ it holds:

$$\mathbb{P}\left[\exists\,\alpha\in\mathbb{F}_q,\,\nu\in\mathcal{R}:\left|\left|A(\nu,\mathbf{t})\cap Z_\alpha(\mathbf{h})\right|-\tfrac{1}{q}\left|A(\nu,\mathbf{t})\right|\right|>\gamma\right] \;\leq\; \frac{q^{k+1/2}}{\gamma^{3/2}}+\iota(\mathbf{h},\mathbf{t})$$

Proof. It obviously suffices to give a proof for the case that $\mathbf{t}$ and $\mathbf{h}$ are statistically independent, i.e. $\iota(\mathbf{h},\mathbf{t})=0$. Now, as $\mathbf{t}$ and $\mathbf{h}$ are independent, we may just assume that w.l.o.g. $\mathbb{P}[\mathbf{t}=t]=1$ for some worst case constant $t\in\mathcal{Q}$. However, once we have fixed $\mathbf{t}$, we can consider α and ν as function values of $\mathbf{h}$, which we denote by $\alpha(\mathbf{h})$ and $\nu_{\mathbf{h}}$ respectively. Thereby, for each $h\in\mathcal{H}$ we can define the equivalence class $[h] := \{h'\in\mathcal{H} \mid A(t,\nu_{h'})=A(t,\nu_h)\}$. Further, let $\bar{\mathcal{H}}\subseteq\mathcal{H}$ denote

a representative system for these equivalence classes, i.e. $\left|\bar{\mathcal{H}} \cap [h]\right| = 1$ for all $h \in \mathcal{H}$. Let some arbitrary $\gamma \in \mathbb{R}_{>0}$ be given. By construction we have:

$$\mathbb{P}\left[\left||A(\nu_{\mathbf{h}},t) \cap Z_{\alpha(\mathbf{h})}(\mathbf{h})| - \tfrac{1}{q}|A(\nu_{\mathbf{h}},t)|\right| > \gamma\right] \;\leq\; \sum_{h\in\bar{\mathcal{H}}} \mathbb{P}\left[\left||A(\nu_h,t) \cap Z_{\alpha(\mathbf{h})}(\mathbf{h})| - \tfrac{1}{q}|A(\nu_h,t)|\right| > \gamma\right]$$

Note that we can discard all summands with $|A(t,\nu_h)| < \gamma$ on the right side, since for any $X \subseteq \mathbb{F}_q^{1\times k}$, $\alpha \in \mathbb{F}_q$, $h \in \mathcal{H}$ it trivially holds that $||X \cap Z_\alpha(h)| - \frac{1}{q}|X|| < |X|$. With $\hat{\mathcal{H}} := \{h \in \bar{\mathcal{H}} \mid \gamma < |A(t,\nu_h)|\}$ we get:

$$\mathbb{P}\left[\left||A(\nu_{\mathbf{h}},t) \cap Z_{\alpha(\mathbf{h})}(\mathbf{h})| - \tfrac{1}{q}|A(\nu_{\mathbf{h}},t)|\right| > \gamma\right] \;\leq\; \sum_{h\in\hat{\mathcal{H}}} \mathbb{P}\left[\left||A(\nu_h,t) \cap Z_{\alpha(\mathbf{h})}(\mathbf{h})| - \tfrac{1}{q}|A(\nu_h,t)|\right| > \gamma\right]$$

However, since $\mathbb{E}(\mathbf{x}) \geq \gamma \cdot \mathbb{P}[\mathbf{x} \geq \gamma]$ for every random variable $\mathbf{x} \in \mathbb{R}$, we can estimate by Lemma 53 for all $\nu \in \mathcal{Q}$:

$$\mathbb{P}\left[\left||A(\nu,t) \cap Z_{\alpha(\mathbf{h})}(\mathbf{h})| - \tfrac{1}{q}|A(\nu,t)|\right| \geq \gamma\right] \;\leq\; \tfrac{1}{\gamma}\sqrt{q \cdot |A(\nu,t)|}$$

It follows:

$$\mathbb{P}\left[\left||A(\nu_{\mathbf{h}},t) \cap Z_{\alpha(\mathbf{h})}(\mathbf{h})| - \tfrac{1}{q}|A(\nu_{\mathbf{h}},t)|\right| > \gamma\right] \;\leq\; \tfrac{1}{\gamma}\sum\nolimits_{h\in\hat{\mathcal{H}}} \sqrt{q \cdot |A(\nu_h,t)|}$$

Since $\|\vec{v}\|_1 \leq \sqrt{m} \cdot \|\vec{v}\|_2$ for all $m \in \mathbb{N}$, $\vec{v} \in \mathbb{R}^m$, we can conclude that $\sum_{i=1}^m |a_i| \leq \sqrt{m \cdot \sum_{i=1}^m a_i^2}$ for any $a_1, \ldots, a_m \in \mathbb{R}$. Thus, it holds:

$$\sum\nolimits_{h\in\hat{\mathcal{H}}} \sqrt{|A(t,\nu_h)|} \;\leq\; \sqrt{|\hat{\mathcal{H}}| \cdot \sum\nolimits_{h\in\hat{\mathcal{H}}} |A(t,\nu_h)|}$$

Since by construction $\{A(t,\nu_h)\}_{h\in\hat{\mathcal{H}}}$ is a disjoint decomposition of some subset of $\mathbb{F}_q^{1\times k}$, we can further estimate:

$$\sum\nolimits_{h\in\hat{\mathcal{H}}} |A(t,\nu_h)| \;\leq\; |\mathbb{F}_q^{1\times k}| \;=\; q^k$$

Further note that by construction $|\hat{\mathcal{H}}| < \frac{1}{\gamma} \cdot q^k$. Putting things together, we have shown:

$$\mathbb{P}\left[\left||A(t,\nu_{\mathbf{h}}) \cap Z_{\alpha(\mathbf{h})}(\mathbf{h})| - \tfrac{1}{q}|A(t,\nu_{\mathbf{h}})|\right| > \gamma\right] \;<\; \frac{q^{k+1/2}}{\gamma^{3/2}} \qquad \square$$

8.3.5 The simulator for a corrupted Goliath

In each choice phase, the simulator for a corrupted Goliath has to extract the correct affine function parameters $(a_i, b_i) \in \mathbb{F}_q^k \times \mathbb{F}_q^k$ and send them to the ideal functionality $\mathcal{F}_{\text{OAFE}}^{\text{seq-ot}}$ (q.v. Figure 6.3). Note that the simulator has no influence on the choice phases (he even is not activated at all), as long as David is not corrupted.

Our simulator for a corrupted Goliath is given in Figure 8.2. The high level picture how this simulator works is as follows. The send phases are simulated straightforwardly: $\mathcal{Z}$ just interacts with a simulated version of the complete real model. When the i-th send phase is over, the simulator must extract a valid Goliath input (a_i, b_i, i), i.e. the simulator needs a description of the token functionality for the i-th choice phase. Therefor, the simulator first checks whether the token acts honestly on random input. If the token's output appears faulty, the simulator henceforth gives default input $(0, 0, i)$ to the ideal functionality; otherwise he rewinds the token to the beginning of the i-th choice phase and inputs other vectors $v_i \in \mathbb{F}_q^{1\times k}$ until he can extract an affine function that describes the token behavior in this phase. Once having extracted this affine description of the token functionality, the simulator can easily compute the unique Goliath input (a_i, b_i, i) corresponding to this token functionality and the messages of the i-th send phase. Note that the running time of $\mathcal{S}^{\text{Goliath}}(\mathcal{A})$ is not a priori polynomially bounded in the security parameter $\lambda := k \log q$, but there may be up to q^k simulated token queries in step 2 of the simulator's extraction program. However, the *expected* number of iterations in that step is constant. We also refer to Section 9.1 for a further discussion on this issue.

Simulator $\mathcal{S}^{\text{Goliath}}(\mathcal{A})$

- Set up an honest David-machine $\mathcal{D}$; also set up a simulated version of $\mathcal{F}_{\text{wrap}}^{\text{stateful}}$ and the given real model adversary $\mathcal{A}$ (which especially impersonates the corrupted Goliath). Wire the simulated machines $\mathcal{A}, \mathcal{D}, \mathcal{F}_{\text{wrap}}^{\text{stateful}}$ to each other and $\mathcal{A}$ to the environment right the way they would be wired in the real model with protocol $\Pi_{\text{OAFE}}^{\text{semi-int}}$ (q.v. Figure 7.1). Further, initialize $f_0 \leftarrow \top$.
- Whenever $\mathcal{D}$ outputs $(\texttt{ready}, i)$, extract a snapshot $\mathcal{T}'$ of $\mathcal{T}$ (including its current internal state) from the view of the simulated $\mathcal{F}_{\text{wrap}}^{\text{stateful}}$ and extract $C, G, \tilde{r}_i, \tilde{S}_i, h_i, \tilde{a}_i, \tilde{b}_i$ from the view of $\mathcal{D}$. Then run the following extraction program:
  1. Pick a random vector $u_i \xleftarrow{\text{r}} \mathbb{F}_q^{1\times k}$ and input (u_i, i) into the token $\mathcal{T}$; let W_i denote the token's output (w.l.o.g. $W_i \in \mathbb{F}_q^{4l\times k}$). If $f_{i-1} = \bot$ or $CW_i \neq \tilde{r}_i u_i + \tilde{S}_i$, set $f_i \leftarrow \bot$ and go to step 4 of this extraction program; else just set $f_i \leftarrow \top$.
  2. Pick a random vector $v_i \xleftarrow{\text{r}} \mathbb{F}_q^{1\times k}$ and input (v_i, i) into a copy of $\mathcal{T}'$; let W' denote the token's output (w.l.o.g. $W' \in \mathbb{F}_q^{4l\times k}$). Retry this step until $CW' = \tilde{r}_i v_i + \tilde{S}_i$ or q^k iterations have past; in the latter case give up, i.e. send $(\star, i)$ to the environment and terminate. If afterwards $u_i = v_i$ or any row of the matrix $W_i - W'$ is linearly independent of $u_i - v_i$, also give up.
  3. Compute the unique vector $r_i \in \mathbb{F}_q^{4l}$, such that $W_i - W' = r_i(u_i - v_i)$, and set $S_i \leftarrow W_i - r_i u_i$. Then compute $a_i \leftarrow \tilde{a}_i + Gr_i$ and $b_i \leftarrow \tilde{b}_i + GS_i h_i$.
  4. If $f_i = \top$, send (a_i, b_i, i) on behalf of the corrupted Goliath to the ideal functionality $\mathcal{F}_{\text{OAFE}}^{\text{seq-ot}}$; else send $(0, 0, i)$.

  Finally, upon receiving $(\texttt{created}, i)$ from the ideal functionality $\mathcal{F}_{\text{OAFE}}^{\text{seq-ot}}$, reply with $(\texttt{Delivery}, i)$.

Figure 8.2: The simulator program $\mathcal{S}^{\text{Goliath}}(\mathcal{A})$, given an adversary $\mathcal{A}$ that corrupts Goliath.

Lemma 55. *Let some arbitrary environment $\mathcal{Z}$ be given and some adversary $\mathcal{A}$ that corrupts the sender Goliath. Then the expected running time of the simulator $\mathcal{S}^{\text{Goliath}}(\mathcal{A})$ is polynomially bounded in the running time of $\mathcal{A}$ and the corresponding token $\mathcal{T}$. In particular, for each simulated send phase the expected number of iterations performed in step 2 of the simulator's extraction program (q.v. Figure 8.2) is constant.*

Proof. When the simulator enters his extraction program, we can express by a variable p the probability that he picks some u_i passing the check in step 1. Then, in each iteration of step 2 with probability $1 - p$ he will pick some v_i that does not pass the check. Hence, if the simulator would not give up after q^k iterations but try on infinitely, we had the following probability that exactly t iterations are performed:

$$\begin{cases} 1 - p & \text{for } t = 0 \\ p^2 \cdot (1-p)^{t-1} & \text{for } t > 0 \end{cases}$$

This yields the following upper bound for the expected number of iterations:

$$p^2 \cdot \sum_{t=1}^{\infty} t \cdot (1-p)^{t-1}$$

Note that w.l.o.g. $p > 0$, as otherwise step 2 of the extraction program is not entered at all. However, if $p > 0$, we can use the well-known formula for the expectation of a geometric distribution:

$$p \cdot \sum_{t=1}^{\infty} t \cdot (1-p)^{t-1} \;=\; \tfrac{1}{p}$$

Putting things together, we have shown that the expected number of iterations is at most 1. □

Functionality $\mathcal{F}'$

Parametrized by a finite vector space $\mathbb{F}_q^k$ and some runtime bound n that is polynomially bounded in the security parameter $\lambda := k \log q$. The counters $j_{\mathsf{created}}, j_{\mathsf{sent}}, j_{\mathsf{queried}}$ are all initialized to 0.

Send phases:

- Upon receiving input (a, b, i) from Goliath, verify that $a, b \in \mathbb{F}_q^k$ and $i = j_{\mathsf{created}} + 1 \leq n$; else ignore that input. Next, update $j_{\mathsf{created}} \leftarrow i$ and send $(\texttt{created}, i)$ to the simulator.
- Upon receiving a message $(\texttt{Delivery}, i)$ from the simulator, verify that $i = j_{\mathsf{sent}} + 1 \leq j_{\mathsf{created}}$; else ignore that message. Next, update $j_{\mathsf{sent}} \leftarrow i$ and send $(\texttt{ready}, i)$ to David.

Choice phases:

- Upon receiving input (x, i) from David, verify that $x \in \mathbb{F}_q$ and $i = j_{\mathsf{queried}} + 1 \leq j_{\mathsf{sent}}$; else ignore that input. Next, update $j_{\mathsf{queried}} \leftarrow i$, send $(\texttt{queried}, x, i)$ to the simulator and wait for the simulator's next message. Then, upon receiving $(\texttt{Reply}, y, i)$ from the simulator, output (y, i) to David.

When a party is corrupted, the simulator is granted unrestricted access to the channel between $\mathcal{F}'$ and the corrupted party, including the ability of deleting and/or forging arbitrary messages.

Figure 8.3: The ideal functionality for the hybrid games $\mathsf{Game}_0, \ldots, \mathsf{Game}_n$. The difference to the ideal functionality $\mathcal{F}_{\text{wrap}}^{\text{stateful}}$ (q.v. Figure 6.3) is that in the choice phases the simulator learns David's input x and may overwrite the respective output y. The inputs a, b in the send phase are just meaningless.

8.3.6 A sequence of hybrid games

We prove indistinguishability between the ideal model and the real model by a hybrid argument. In particular, we will show that for $l = 1, \ldots, n$ no environment can distinguish non-negligibly between some hybrid games Game_{l-1} and Game_l, where Game_0 and Game_n are indistinguishable from the ideal and real model respectively. Each hybrid game Game_l works like an ideal model with ideal functionality $\mathcal{F}'$ and (non-efficient) simulator $\mathcal{S}_l'(\mathcal{A})$. The functionality $\mathcal{F}'$ resembles the ideal functionality $\mathcal{F}_{\text{OAFE}}^{\text{seq-ot}}$, but the simulator learns the ideal David's inputs and may overwrite the corresponding outputs of $\mathcal{F}'$. For a formal description see Figure 8.3. Each simulator $\mathcal{S}_l'(\mathcal{A})$ overwrites the first l outputs, so that they exactly equal the first l David outputs in the real model. The remaining $n - l$ outputs are computed from an extracted affine description of the token functionality, very similar to the ideal model. For a formal description of the simulators $\mathcal{S}_l'(\mathcal{A})$ see Figure 8.4.

Corollary 56. *Given any adversary $\mathcal{A}$ that corrupts the sender party Goliath, our hybrid game Game_0 with simulator $\mathcal{S}_0'(\mathcal{A})$ is statistically indistinguishable from the ideal model with simulator $\mathcal{S}^{\text{Goliath}}(\mathcal{A})$, and our hybrid game Game_n with simulator $\mathcal{S}_n'(\mathcal{A})$ is perfectly indistinguishable from the real model with adversary $\mathcal{A}$.*

Proof. It is straightforward to see that Game_n is perfectly indistinguishable from the real model. It is also straightforward to see that Game_0 is perfectly indistinguishable from the ideal model conditioned to the event that the simulator $\mathcal{S}^{\text{Goliath}}(\mathcal{A})$ does not reach the iteration bound q^k in step 2 of his extraction program. However, by Lemma 55 this iteration bound is only reached with negligible probability and thus Game_0 is statistically indistinguishable from the ideal model. □

8.3.7 Transformation of successive hybrid games

For our security proof we have to show that from the environment's view any successive hybrid games $\mathsf{Game}_{l-1}, \mathsf{Game}_l$, parametrized with the finite vector space $\mathbb{F}_q^k$ and runtime bound n, are

Simulator $\mathcal{S}'_l(\mathcal{A})$

- Set up an honest David-machine $\mathcal{D}$; also set up a simulated version of $\mathcal{F}_{\text{wrap}}^{\text{stateful}}$ and the given real model adversary $\mathcal{A}$ (which especially impersonates the corrupted Goliath). Wire the simulated machines $\mathcal{A}, \mathcal{D}, \mathcal{F}_{\text{wrap}}^{\text{stateful}}$ to each other and $\mathcal{A}$ to the environment right the way they would be wired in the real model. Further, initialize $f_0 \leftarrow \top$.
- Whenever $\mathcal{D}$ outputs $(\texttt{ready}, i)$, choose any $a, b \in \mathbb{F}_q^k$ and send (a, b, i) on behalf of the corrupted Goliath to the functionality $\mathcal{F}'$. Then, upon receiving $(\texttt{created}, i)$ from $\mathcal{F}'$, reply with $(\texttt{Delivery}, i)$.
- Upon receiving $(\texttt{queried}, x_i, i)$ from the functionality $\mathcal{F}'$, extract the token function τ_i of the current choice phase from the view of the simulated $\mathcal{F}_{\text{wrap}}^{\text{stateful}}$, in the sense that on input $(\tilde{v}, i)$ the token $\mathcal{T}$ currently would output $\tau_i(\tilde{v})$. If $\tau_i(\tilde{v}) \notin \mathbb{F}_q^{4k \times k}$ for some $\tilde{v} \in \mathbb{F}_q^{1 \times k}$, treat this as an encoding of the all-zero matrix; thus w.l.o.g. $\tau_i : \mathbb{F}_q^{1 \times k} \to \mathbb{F}_q^{4k \times k}$. Further, extract $C, G, \tilde{r}_i, \tilde{S}_i, h_i, \tilde{a}_i, \tilde{b}_i$ from the view of $\mathcal{D}$. Then, if $i > l$, run the following extraction program:
  1. Pick randomly $u_i \xleftarrow{\text{r}} \mathbb{F}_q^{1 \times k}$ and input (u_i, i) into the token $\mathcal{T}$, thus progressing its internal state.
  2. If $f_{i-1} = \top$ and $C \cdot \tau_i(u_i) = \tilde{r}_i u_i + \tilde{S}_i$, set $f_i \leftarrow \top$; otherwise set $f_i \leftarrow \bot$.
  3. If $f_i = \top$, pick a random vector $v_i \xleftarrow{\text{r}} \left\{\tilde{v} \in \mathbb{F}_q^{1 \times k} \;\middle|\; C \cdot \tau_i(\tilde{v}) = \tilde{r}_i \tilde{v} + \tilde{S}_i\right\}$. Then, if $v_i = u_i$ or any row of the matrix $\tau_i(u_i) - \tau_i(v_i)$ is linearly independent of $u_i - v_i$, send $(\star, i)$ to the environment and terminate; else compute $y_i \leftarrow (\tilde{a}_i + G r_i) x_i + \tilde{b}_i + G S_i h_i$, where $r_i \in \mathbb{F}_q^{4k}$ is the unique vector with $\tau_i(u_i) - \tau_i(v_i) = r_i \cdot (u_i - v_i)$, and $S_i := \tau_i(u_i) - r_i u_i$.

     If $f_i = \bot$, just set $y_i \leftarrow 0$ (such that $y_i \in \mathbb{F}_q^k$).
  4. Send $(\texttt{Reply}, y_i, i)$ to $\mathcal{F}'$.

  If $i \leq l$, just run the following program instead:
  1. Pick a random vector $z_i \xleftarrow{\text{r}} \{\tilde{z} \in \mathbb{F}_q^{1 \times k} \mid \tilde{z} h_i = x_i\}$ and input (z_i, i) into the token $\mathcal{T}$, thus progressing its internal state.
  2. If $f_{i-1} = \top$ and $C \cdot \tau_i(z_i) = \tilde{r}_i z_i + \tilde{S}_i$, set $f_i \leftarrow \top$; otherwise set $f_i \leftarrow \bot$.
  3. If $f_i = \top$, compute $y_i \leftarrow G \cdot \tau_i(z_i) \cdot h_i + \tilde{a}_i x_i + \tilde{b}_i$.

     If $f_i = \bot$, just set $y_i \leftarrow 0$ (such that $y_i \in \mathbb{F}_q^k$).
  4. Send $(\texttt{Reply}, y_i, i)$ to $\mathcal{F}'$.

Figure 8.4: The simulator program $\mathcal{S}'_l(\mathcal{A})$ for the hybrid game Game_l, given an adversary $\mathcal{A}$ that corrupts the sender Goliath. The hybrid games Game_n and Game_0 are indistinguishable from the real model with adversary $\mathcal{A}$ and the ideal model with simulator $\mathcal{S}^{\text{Goliath}}(\mathcal{A})$ respectively.

statistically indistinguishable. Our approach is to transform these hybrid games into an indistinguishability game $\Gamma_0(\mathbb{F}_q, n, l)$, so that every environment $\mathcal{Z}$ that can distinguish Game_{l-1} from Game_l corresponds to a player that wins in $\Gamma_0(\mathbb{F}_q, n, l)$ with some non-negligible advantage. This approach allows us to successively modify the obtained indistinguishability game $\Gamma_0(\mathbb{F}_q, n, l)$, so that it becomes feasible to derive a maximum winning probability from which we can then infer a negligible upper bound for the statistical distance between $\mathcal{Z}$'s views in Game_{l-1} and Game_l. The intuition behind this sequence of indistinguishability games can be sketched as follows.

$\Gamma_0(\mathbb{F}_q, n, l)$: This is just a straightforward reformulation of what the environment $\mathcal{Z}$ sees in the hybrid games Game_{l-1} and Game_l respectively.

$\Gamma_1(\mathbb{F}_q, n, l)$: We make the player a bit stronger by giving him more direct access to the internal game state.

$\Gamma_2(\mathbb{F}_q, n, l, \varepsilon)$: We exploit that the token is somehow committed to affine behavior (cf. Section 8.3.2). This allows us to unify the way, David's outputs are computed in the hybrid real part and the hybrid ideal part: Basically, David's outputs in the hybrid real part are now also computed from an extracted affine approximation of the token functionality. The additional game

parameter ε is introduced for technical reasons; it will be needed later to apply Lemma 51.

$\Gamma_3(\mathbb{F}_q, n, l, \varepsilon)$**:** We replace the simulator's abort message $(\star, i)$, q.v. step 2 in Figure 8.2. This corresponds to a simulator modification, so that he may not give up any more, but instead switches to the mode where David henceforth produces default (all-zero) output.

$\Gamma_4(\mathbb{F}_q, n, l, \varepsilon)$**:** We exploit that the token functionality for most inputs can be approximated by no more than one affine function (cf. Section 8.3.3). This allows us to consider the extracted affine function parameters as token outputs rather than approximations of the token functionality.

$\Gamma_5(\mathbb{F}_q, n, l)$**:** We no longer only *consider* the extracted affine function parameters as token outputs; now they are.

$\Gamma_6(\mathbb{F}_q, n, l)$**:** We make the player stronger. We let him learn the first $l-1$ token inputs and let him choose the last $n-l$ token inputs. Thus only the stage l, which is the only stage in which Game_{l-1} differs from Game_l, stays out of control of the player.

$\Gamma_7(\mathbb{F}_q, n, l)$**:** We just exploit that several variables have become obsolete, and get rid of them.

$\Gamma_8(\mathbb{F}_q, n, l)$**:** We get rid of the challenge matrix C. The player must now exactly forecast token outputs rather than only linear projections.

$\Gamma_9(\mathbb{F}_q, n, l)$**:** We make it explicit that w.l.o.g. the player follows a deterministic strategy, solely depending on what he learns during the game run. This is the final version of our indistinguishability game.

We give now a detailed description of our indistinguishability game (Figure 8.5) and its relation to Game_{l-1} and Game_l (Lemma 57). Then we successively transform it and show how this affects the winning probability.

Lemma 57. *Let $\frac{1}{2}+\delta$ be the maximal winning probability in the game $\Gamma_0(\mathbb{F}_q^k, n, l)$. Then the statistical distance between the environment's view in Game_{l-1} and its view in Game_l, both parametrized with the finite vector space $\mathbb{F}_q^k$ and runtime bound n, is upper bounded by 2δ.*

Proof. The proof is absolutely straightforward. Basically, we just have to show how the player $\mathcal{K}$ in $\Gamma_0(\mathbb{F}_q^k, n, l)$ can perfectly emulate the hybrid game Game_{l-d} for an environment $\mathcal{Z}$, where d is the secret challenge bit that $\mathcal{K}$ finally tries to guess. Our player $\mathcal{K}$ just works as follows:

- Setup a simulated version of the given environment $\mathcal{Z}$ and the complete hybrid game Game_l.
- As soon as in the game Game_l the token $\mathcal{T}$ is fixed, specify the mappings $\tau_1, \ldots, \tau_n$ such that the token functionality in the i-th choice phase on input history $(w_1, 1), \ldots, (w_{i-1}, i-1)$ implements the function $(w_i, i) \mapsto \tau_i(w_1, \ldots, w_i)$. This is step 1 of $\Gamma_0(\mathbb{F}_q^k, n, l)$.
- Always overwrite the simulated David's random choices of $C, G, h_1, \ldots, h_n$ by the respective values learned in step 2 of $\Gamma_0(\mathbb{F}_q^k, n, l)$.
- Whenever some $\tilde{r}_i, \tilde{S}_i, \tilde{a}_i, \tilde{b}_i$ are to be chosen in step 4a or step 5a of $\Gamma_0(\mathbb{F}_q^k, n, l)$, just take the respective values from the simulated David's view. Analogously take x_i from the view of the simulated functionality $\mathcal{F}'$.
- Whenever the simulated functionality $\mathcal{F}'$ outputs some (y_i, i), overwrite y_i by the respective value learned in step 4c or step 5c of $\Gamma_0(\mathbb{F}_q^k, n, l)$ respectively.
- Upon receiving a special message $(\star, i)$, just forward it to the simulated environment $\mathcal{Z}$ and stop the simulated hybrid game.

Game $\Gamma_0(\mathbb{F}_q^k, n, l)$

Parametrized by a finite vector space $\mathbb{F}_q^k$ and some $n, l \in \mathbb{N}_{>0}$ with $l \leq n$. Let $\mathcal{U} := \mathbb{F}_q^{1\times k}$ and $\mathcal{H} := \mathbb{F}_q^k \setminus \{0\}$. The player $\mathcal{K}$ is computationally unbounded.

1. For each $i \in \{1, \ldots, n\}$ the player $\mathcal{K}$ specifies some mapping $\tau_i : \mathcal{U}^i \to \mathbb{F}_q^{4k\times k}$.
2. The player $\mathcal{K}$ learns n random vectors $h_1, \ldots, h_n \xleftarrow{\text{r}} \mathcal{H}$, a random matrix $C \xleftarrow{\text{r}} \mathbb{F}_q^{3k\times 4k}$ and some $G \in \mathbb{F}_q^{k\times 4k}$ complementary to C.
3. A challenge bit $d \xleftarrow{\text{r}} \{0, 1\}$ is chosen secretly.
4. For $i = 1, \ldots, l - d$:
   a) The player $\mathcal{K}$ chooses some $\tilde{r}_i \in \mathbb{F}_q^{3k}$, $\tilde{S}_i \in \mathbb{F}_q^{3k\times k}$, $\tilde{a}_i, \tilde{b}_i \in \mathbb{F}_q^k$ and $x_i \in \mathbb{F}_q$.
   b) Let $w_i := z_i \xleftarrow{\text{r}} \{\tilde{z} \in \mathcal{U} \mid \tilde{z}h_i = x_i\}$, chosen secretly.
   c) If $C \cdot \tau_j(w_1, \ldots, w_j) \neq \tilde{r}_j w_j + \tilde{S}_j$ for any $j \leq i$, the player $\mathcal{K}$ learns $y_i := 0 \in \mathbb{F}_q^k$.
   Otherwise, $\mathcal{K}$ learns $y_i := G \cdot \tau_i(w_1, \ldots, w_i) \cdot h_i + \tilde{a}_i x_i + \tilde{b}_i$.
5. For $i = l - d + 1, \ldots, n$:
   a) The player $\mathcal{K}$ chooses some $\tilde{r}_i \in \mathbb{F}_q^{3k}$, $\tilde{S}_i \in \mathbb{F}_q^{3k\times k}$, $\tilde{a}_i, \tilde{b}_i \in \mathbb{F}_q^k$ and $x_i \in \mathbb{F}_q$.
   b) Let $w_i := u_i \xleftarrow{\text{r}} \mathcal{U}$, chosen secretly.
   c) If $C \cdot \tau_j(w_1, \ldots, w_j) \neq \tilde{r}_j w_j + \tilde{S}_j$ for any $j \leq i$, the player $\mathcal{K}$ learns $y_i := 0 \in \mathbb{F}_q^k$.
   Otherwise, a random vector $v_i \xleftarrow{\text{r}} \left\{\tilde{v} \in \mathcal{U} \;\middle|\; C \cdot \tau_i(w_1, \ldots, w_{i-1}, \tilde{v}) = \tilde{r}_i\tilde{v} + \tilde{S}_i\right\}$ is chosen secretly. Then, if $v_i = w_i$ or any row of the matrix $\tau_i(w_1, \ldots, w_i) - \tau_i(w_1, \ldots, w_{i-1}, v_i)$ is linearly independent of $w_i - v_i$, the player $\mathcal{K}$ receives a special message $(\star, i)$ and the game is aborted in the sense that step 6 follows next; else $\mathcal{K}$ learns $y_i := (\tilde{a}_i + Gr_i)x_i + \tilde{b}_i + GS_ih_i$, where $r_i \in \mathbb{F}_q^{4k}$ is the unique vector with $\tau_i(w_1, \ldots, w_i) - \tau_i(w_1, \ldots, w_{i-1}, v_i) = r(w_i - v_i)$, and $S_i := \tau_i(w_1, \ldots, w_i) - r_i w_i$.
6. The player $\mathcal{K}$ computes and outputs a guess bit $\tilde{d} \in \{0, 1\}$. He wins the game, if $\tilde{d} = d$.

Figure 8.5: Definition of a stand-alone indistinguishability game that captures the difference between the hybrid games Game_{l-1} and Game_l. The player's view in the indistinguishability game $\Gamma_0(\mathbb{F}_q^k, n, l)$ corresponds straightforwardly to the environment's view in the hybrid game Game_{l-d}, where d is the secret challenge bit from step 3 of $\Gamma_0(\mathbb{F}_q^k, n, l)$. Thus, the statistical distance between the its view in Game_{l-1} and the environment's view in Game_l is upper bounded by 2δ, where $\frac{1}{2} + \delta$ is the maximum winning probability in the indistinguishability game $\Gamma_0(\mathbb{F}_q^k, n, l)$.

It is straightforward to see that this way the player $\mathcal{K}$ perfectly emulates a view of $\mathcal{Z}$ in the hybrid game Game_{l-d}; this is just how we constructed the game $\Gamma_0(\mathbb{F}_q^k, n, l)$.

Now, let the random variable $\mathbf{view}_{\mathcal{Z}}$ denote this emulated view of $\mathcal{Z}$, and let the random variable $\mathbf{d}$ denote the secret challenge bit that $\mathcal{K}$ tries to guess in step 6 of $\Gamma_0(\mathbb{F}_q^k, n, l)$. Since by assumption the player $\mathcal{K}$ wins the game $\Gamma_0(\mathbb{F}_q^k, n, l)$ at most with probability $\frac{1}{2} + \delta$, it must hold for every predicate P that $\mathbb{P}\big[P(\mathbf{view}_{\mathcal{Z}}) = 0 \;\wedge\; \mathbf{d} = 0\big] + \mathbb{P}\big[P(\mathbf{view}_{\mathcal{Z}}) = 1 \;\wedge\; \mathbf{d} = 1\big] \leq \frac{1}{2} + \delta$. Furthermore, note that there does exist a predicate P such that we can write the statistical distance $dist_l$ between the views of $\mathcal{Z}$ in Game_{l-1} and Game_l as follows:

$$dist_l \;=\; \overbrace{\mathbb{P}\big[P(\mathbf{view}_{\mathcal{Z}}) = 1 \;\big|\; \mathbf{d} = 1\big]}^{=\mathbb{P}[P(\text{view of } \mathcal{Z} \text{ in } \mathsf{Game}_{l-1})=1]} - \overbrace{\mathbb{P}\big[P(\mathbf{view}_{\mathcal{Z}}) = 1 \;\big|\; \mathbf{d} = 0\big]}^{=\mathbb{P}[P(\text{view of } \mathcal{Z} \text{ in } \mathsf{Game}_{l})=1]}$$

Thus we can conclude:

$$dist_l \;=\; \underbrace{\mathbb{P}\big[P(\mathbf{view}_{\mathcal{Z}}) = 1 \;\big|\; \mathbf{d} = 1\big]}_{=2\cdot\mathbb{P}[P(\mathbf{view}_{\mathcal{Z}})=1\,\wedge\,\mathbf{d}=1]} - \Big(1 - \underbrace{\mathbb{P}\big[P(\mathbf{view}_{\mathcal{Z}}) = 0 \;\big|\; \mathbf{d} = 0\big]}_{=2\cdot\mathbb{P}[P(\mathbf{view}_{\mathcal{Z}})=0\,\wedge\,\mathbf{d}=0]}\Big) \;\leq\; 2\delta \qquad \square$$

Game $\Gamma_1(\mathbb{F}_q^k, n, l)$

Parametrized by a finite vector space $\mathbb{F}_q^k$ and some $n, l \in \mathbb{N}_{>0}$ with $l \leq n$. Let $\mathcal{U} := \mathbb{F}_q^{1\times k}$ and $\mathcal{H} := \mathbb{F}_q^k \setminus \{0\}$. The player $\mathcal{K}$ is computationally unbounded.

1. For each $i \in \{1, \ldots, n\}$ the player $\mathcal{K}$ specifies some mapping $\tau_i : \mathcal{U}^i \to \mathbb{F}_q^{4k\times k}$.
2. The player $\mathcal{K}$ learns n random vectors $h_1, \ldots, h_n \xleftarrow{r} \mathcal{H}$ and a random matrix $C \xleftarrow{r} \mathbb{F}_q^{3k\times 4k}$.
3. A challenge bit $d \xleftarrow{r} \{0, 1\}$ is chosen secretly.
4. For $i = 1, \ldots, l - d$:
   a) The player $\mathcal{K}$ chooses some $\tilde{r}_i \in \mathbb{F}_q^{3k}$, $\tilde{S}_i \in \mathbb{F}_q^{3k\times k}$, $x_i \in \mathbb{F}_q$.
   b) Let $w_i := z_i \xleftarrow{r} \{\tilde{z} \in \mathcal{U} \mid \tilde{z}h_i = x_i\}$, chosen secretly.
   c) If $C \cdot \tau_i(w_1, \ldots, w_i) \neq \tilde{r}_i w_i + \tilde{S}_i$, the player $\mathcal{K}$ receives a special message $(\perp, i)$ and the game is aborted in the sense that step 6 follows next.

      Otherwise, the player $\mathcal{K}$ learns $\tau_i(w_1, \ldots, w_i) \cdot h_i$.
5. For $i = l - d + 1, \ldots, n$:
   a) The player $\mathcal{K}$ chooses some $\tilde{r}_i \in \mathbb{F}_q^{3k}$, $\tilde{S}_i \in \mathbb{F}_q^{3k\times k}$, $x_i \in \mathbb{F}_q$.
   b) Let $w_i := u_i \xleftarrow{r} \mathcal{U}$, chosen secretly.
   c) If $C \cdot \tau_i(w_1, \ldots, w_i) \neq \tilde{r}_i w_i + \tilde{S}_i$, the player $\mathcal{K}$ receives a special message $(\perp, i)$ and the game is aborted in the sense that step 6 follows next.

      Otherwise, a random vector $v_i \xleftarrow{r} \left\{\tilde{v} \in \mathcal{U} \mid C \cdot \tau_i(w_1, \ldots, w_{i-1}, \tilde{v}) = \tilde{r}_i\tilde{v} + \tilde{S}_i\right\}$ is chosen secretly. Then, if $v_i = w_i$ or any row of the matrix $\tau_i(w_1, \ldots, w_i) - \tau_i(w_1, \ldots, w_{i-1}, v_i)$ is linearly independent of $w_i - v_i$, the player $\mathcal{K}$ receives a special message $(\star, i)$ and the game is aborted in the sense that step 6 follows next; else $\mathcal{K}$ learns $r_i x_i + S_i h_i$, where $r_i \in \mathbb{F}_q^{4k}$ is the unique vector with $\tau_i(w_1, \ldots, w_i) - \tau_i(w_1, \ldots, w_{i-1}, v_i) = r(w_i - v_i)$, and $S_i := \tau_i(w_1, \ldots, w_i) - r_i w_i$.
6. The player $\mathcal{K}$ computes and outputs a guess bit $\tilde{d} \in \{0, 1\}$. He wins the game, if $\tilde{d} = d$.

Figure 8.6: First transformation of our stand-alone indistinguishability game. There are two differences to the game $\Gamma_0(\mathbb{F}_q^k, n, l)$. Firstly, where $\Gamma_0(\mathbb{F}_q^k, n, l)$ in step 4c or step 5c switched to a mode such that the player $\mathcal{K}$ henceforth only receives all-zero outputs, now $\mathcal{K}$ is notified about that by a special message $(\perp, i)$ and the game is aborted. Secondly, $\mathcal{K}$ now directly learns $\tau_i(w_1, \ldots, w_i) \cdot h_i$ in step 4c and $r_i x_i + S_i h_i$ in step 5c instead of the corresponding image of $\vartheta \mapsto G\vartheta + \tilde{a}_i x_i + \tilde{b}_i$. These game modifications just make $\mathcal{K}$ strictly stronger and $G, (\tilde{a}_1, \tilde{b}_1), \ldots, (\tilde{a}_n, \tilde{b}_n)$ obsolete.

Lemma 58. *The maximum winning probability in the game $\Gamma_0(\mathbb{F}_q^k, n, l)$ is upper bounded by the maximum winning probability in the game $\Gamma_1(\mathbb{F}_q^k, n, l)$.*

Proof. This holds trivially, since the player in $\Gamma_1(\mathbb{F}_q^k, n, l)$ is strictly stronger than in $\Gamma_0(\mathbb{F}_q^k, n, l)$. □

Lemma 59. *The probability that the game $\Gamma_2(\mathbb{F}_q^k, n, l, \varepsilon)$ is aborted in step 4d, is upper bounded by:*

$$n \cdot (q^{1-(1/3-\varepsilon)k} + q^{1-k} + q^{2-k}) + \sqrt{\exp(n \cdot q^{2-k}) - 1}$$

Proof. Let some arbitrary player $\mathcal{K}$ be given and let the random variables $\mathbf{C}, \mathbf{w}_1, \ldots, \mathbf{w}_n$ represent the same-named values in the game $\Gamma_2(\mathbb{F}_q^k, n, l, \varepsilon)$. It is straightforward to see that for each stage $i \in \{1, \ldots, n\}$ the case 4(d)i occurs at most with probability $p := q^{(2/3+\varepsilon)k}/q^{k-1}$. Further, for each stage $i \in \{1, \ldots, n\}$ we have by Lemma 50 that the case 4(d)ii occurs at most with probability $p' := (q^{1-3k} + q^{2-3k}) \cdot |\mathcal{U}|^2$, if $\mathbf{C}$ and $(\mathbf{w}_1, \ldots, \mathbf{w}_{i-1})$ are statistically independent. Thus, by the Union Bound we can estimate the overall probability that the game $\Gamma_2(\mathbb{F}_q^k, n, l, \varepsilon)$ is aborted in

Game $\Gamma_2(\mathbb{F}_q^k, n, l, \varepsilon)$

Parametrized by a finite vector space $\mathbb{F}_q^k$, some $n, l \in \mathbb{N}_{>0}$ with $l \leq n$, and $\varepsilon \in \mathbb{R}_{>0}$ such that $q^{(2/3+\varepsilon)k} \geq q$. Let $\mathcal{U} := \mathbb{F}_q^{1\times k}$ and $\mathcal{H} := \mathbb{F}_q^k \setminus \{0\}$. The player $\mathcal{K}$ is computationally unbounded.

1. For each $i \in \{1, \ldots, n\}$ the player $\mathcal{K}$ specifies some mapping $\tau_i : \mathcal{U}^i \to \mathbb{F}_q^{4k\times k}$.
2. The player $\mathcal{K}$ learns n random vectors $h_1, \ldots, h_n \xleftarrow{r} \mathcal{H}$ and a random matrix $C \xleftarrow{r} \mathbb{F}_q^{3k\times 4k}$.
3. A challenge bit $d \xleftarrow{r} \{0, 1\}$ is chosen secretly.
4. For $i = 1, \ldots, n$:
   a) The player $\mathcal{K}$ chooses some $\tilde{r}_i \in \mathbb{F}_q^{3k}$, $\tilde{S}_i \in \mathbb{F}_q^{3k\times k}$, $x_i \in \mathbb{F}_q$.
   b) If $i \leq l - d$, let $w_i := z_i \xleftarrow{r} \{\tilde{z} \in \mathcal{U} \mid \tilde{z}h_i = x_i\}$, else let $w_i := u_i \xleftarrow{r} \mathcal{U}$, chosen secretly.
   c) If $C \cdot \tau_i(w_1, \ldots, w_i) \neq \tilde{r}_i w_i + \tilde{S}_i$, the player $\mathcal{K}$ receives a special message $(\perp, i)$ and the game is aborted in the sense that step 5 follows next.
   d) In the following cases the player $\mathcal{K}$ receives a special message $(\star, i)$ and the game is aborted in the sense that step 5 follows next:
      i. $C \cdot \tau_i(w_1, \ldots, w_i) = \tilde{r}_i w_i + \tilde{S}_i$ and $\#\{\tilde{v} \in \mathcal{U} \mid C \cdot \tau_i(w_1, \ldots, w_{i-1}, \tilde{v}) = \tilde{r}_i\tilde{v} + \tilde{S}_i\} \leq q^{(2/3+\varepsilon)k}$.
      ii. There exist some $W, W' \in \tau_i(w_1, \ldots, w_{i-1}, \mathcal{U})$, such that $\operatorname{rank}(CW - CW') \leq 1$ and $\operatorname{rank}(W - W') > \operatorname{rank}(CW - CW')$.

   Otherwise, $\mathcal{K}$ learns $r_i x_i + S_i h_i$, where $(r_i, S_i) \in \mathbb{F}_q^{4k} \times \mathbb{F}_q^{4k\times k}$ is the unique tuple such that $\tau_i(w_1, \ldots, w_{i-1}, v) = r_i v + S_i$ for all $v \in \{\tilde{v} \in \mathcal{U} \mid C \cdot \tau_i(w_1, \ldots, w_{i-1}, \tilde{v}) = \tilde{r}_i\tilde{v} + \tilde{S}_i\}$.
5. The player $\mathcal{K}$ computes and outputs a guess bit $\tilde{d} \in \{0, 1\}$. He wins the game, if $\tilde{d} = d$.

Figure 8.7: Second transformation of our stand-alone indistinguishability game. The difference to the game $\Gamma_1(\mathbb{F}_q^k, n, l)$ is the now uniform way to compute outputs for $\mathcal{K}$. Note that the tuple (r_i, S_i) in step 4d is well-defined by Lemma 49.

step 4d by $n \cdot (p + p') + \iota(\mathbf{C}, (\mathbf{w}_1, \ldots, \mathbf{w}_n))$. Estimating $\iota(\mathbf{C}, (\mathbf{w}_1, \ldots, \mathbf{w}_n))$ by Corollary 48 yields:

$$n \cdot (p + p') + \iota(\mathbf{C}, (\mathbf{w}_1, \ldots, \mathbf{w}_n)) \;<\; n \cdot (q^{1-(1/3-\varepsilon)k} + q^{1-k} + q^{2-k}) + \sqrt{\exp(n \cdot q^{2-k}) - 1} \qquad \square$$

Lemma 60. *The statistical distance between $\mathcal{K}$'s view in the game $\Gamma_1(\mathbb{F}_q^k, n, l)$ and $\mathcal{K}$'s view in the game $\Gamma_2(\mathbb{F}_q^k, n, l, \varepsilon)$ is upper bounded by:*

$$n \cdot (q^{-(2/3+\varepsilon)k} + q^{1-(1/3-\varepsilon)k} + q^{1-k} + q^{2-k}) + \sqrt{\exp(n \cdot q^{2-k}) - 1}$$

Proof. Let some arbitrary player $\mathcal{K}$ for the game $\Gamma_2(\mathbb{F}_q^k, n, l, \varepsilon)$ be given. Note that the only difference to the game $\Gamma_1(\mathbb{F}_q^k, n, l)$ is the computation of $\mathcal{K}$'s output in step 4d, which is now the same for $i \leq l - d$ and $i > l - d$. Since by Lemma 59 we already have an estimation for the abort probability in step 4d, it suffices to consider the case that $\mathcal{K}$ actually learns $r_i x_i + S_i h_i$. If $i \leq l - d$, we can just argue that $r_i x_i + S_i h_i = (r_i w_i + S_i)h_i = \tau_i(w_1, \ldots, w_i) \cdot h_i$ by construction, and thus the player $\mathcal{K}$ receives exactly the same as he would have received in step 4c of $\Gamma_1(\mathbb{F}_q^k, n, l)$. For $i > l - d$, we exploit the following facts:

- If the game is not aborted afore, in step 4d of $\Gamma_2(\mathbb{F}_q^k, n, l, \varepsilon)$ the player $\mathcal{K}$ learns $r_i x_i + S_i h_i$, where $(r_i, S_i) \in \mathbb{F}_q^{4k} \times \mathbb{F}_q^{4k\times k}$ is the unique tuple such that $\tau_i(w_1, \ldots, w_{i-1}, v) = r_i v + S_i$ for all $v \in \{\tilde{v} \in \mathcal{U} \mid C \cdot \tau_i(w_1, \ldots, w_{i-1}, \tilde{v}) = \tilde{r}_i\tilde{v} + \tilde{S}_i\}$.
- Thus, for every $v \in \{\tilde{v} \in \mathcal{U} \mid C \cdot \tau_i(w_1, \ldots, w_{i-1}, \tilde{v}) = \tilde{r}_i\tilde{v} + \tilde{S}_i\}$ either $v = w_i$, or r_i is the unique vector with $\tau_i(w_1, \ldots, w_i) - \tau_i(w_1, \ldots, w_{i-1}, v_i) = r(w_i - v_i)$ and it holds that $S_i = \tau_i(w_1, \ldots, w_i) - r_i w_i$.

Game $\Gamma_3(\mathbb{F}_q^k, n, l, \varepsilon)$

Parametrized by a finite vector space $\mathbb{F}_q^k$, some $n, l \in \mathbb{N}_{>0}$ with $l \leq n$, and $\varepsilon \in \mathbb{R}_{>0}$ such that $q^{(2/3+\varepsilon)k} \geq q$. Let $\mathcal{U} := \mathbb{F}_q^{1\times k}$ and $\mathcal{H} := \mathbb{F}_q^k \setminus \{0\}$. The player $\mathcal{K}$ is computationally unbounded.

1. For each $i \in \{1, \ldots, n\}$ the player $\mathcal{K}$ specifies some mapping $\tau_i : \mathcal{U}^i \to \mathbb{F}_q^{4k\times k}$.
2. The player $\mathcal{K}$ learns n random vectors $h_1, \ldots, h_n \xleftarrow{\text{r}} \mathcal{H}$ and a random matrix $C \xleftarrow{\text{r}} \mathbb{F}_q^{3k\times 4k}$.
3. A challenge bit $d \xleftarrow{\text{r}} \{0, 1\}$ is chosen secretly.
4. For $i = 1, \ldots, n$:
   a) The player $\mathcal{K}$ chooses some $\tilde{r}_i \in \mathbb{F}_q^{3k}$, $\tilde{S}_i \in \mathbb{F}_q^{3k\times k}$, $x_i \in \mathbb{F}_q$.
   b) If $i \leq l - d$, let $w_i := z_i \xleftarrow{\text{r}} \{\tilde{z} \in \mathcal{U} \mid \tilde{z}h_i = x_i\}$, else let $w_i := u_i \xleftarrow{\text{r}} \mathcal{U}$, chosen secretly.
   c) In the following cases the player $\mathcal{K}$ receives a special message $(\perp, i)$ and the game is aborted in the sense that step 5 follows next:
      i. It holds that $C \cdot \tau_i(w_1, \ldots, w_i) \neq \tilde{r}_i w_i + \tilde{S}_i$.
      ii. It holds that $\#\{\tilde{v} \in \mathcal{U} \mid C \cdot \tau_i(w_1, \ldots, w_{i-1}, \tilde{v}) = \tilde{r}_i\tilde{v} + \tilde{S}_i\} \leq q^{(2/3+\varepsilon)k}$.
      iii. There exist some $W, W' \in \tau_i(w_1, \ldots, w_{i-1}, \mathcal{U})$, such that $\operatorname{rank}(CW - CW') \leq 1$ and $\operatorname{rank}(W - W') > \operatorname{rank}(CW - CW')$.

      Otherwise, $\mathcal{K}$ learns $r_i x_i + S_i h_i$, where $(r_i, S_i) \in \mathbb{F}_q^{4k} \times \mathbb{F}_q^{4k\times k}$ is the unique tuple such that $\tau_i(w_1, \ldots, w_{i-1}, v) = r_i v + S_i$ for all $v \in \{\tilde{v} \in \mathcal{U} \mid C \cdot \tau_i(w_1, \ldots, w_{i-1}, \tilde{v}) = \tilde{r}_i\tilde{v} + \tilde{S}_i\}$.
5. The player $\mathcal{K}$ computes and outputs a guess bit $\tilde{d} \in \{0, 1\}$. He wins the game, if $\tilde{d} = d$.

Figure 8.8: Third transformation of our stand-alone indistinguishability game. The only difference to the game $\Gamma_2(\mathbb{F}_q^k, n, l, \varepsilon)$ is that the abort message $(\star, i)$ was replaced by $(\perp, i)$.

- Moreover, since $\#\{\tilde{v} \in \mathcal{U} \mid C \cdot \tau_i(w_1, \ldots, w_{i-1}, \tilde{v}) = \tilde{r}_i\tilde{v} + \tilde{S}_i\} > q^{(2/3+\varepsilon)k}$ for non-aborted stages, a uniformly random v may equal w_i with probability less than $q^{-(2/3+\varepsilon)k}$.

Putting these three observations together, we can conclude that with probability higher than $q^{-(2/3+\varepsilon)k}$ in step 5c of $\Gamma_1(\mathbb{F}_q^k, n, l)$ the same tuple (r_i, S_i) and hence the same output $r_i x_i + S_i h_i$ would be generated as in step 4d of $\Gamma_2(\mathbb{F}_q^k, n, l, \varepsilon)$. Thus, conditioned to the event that $\Gamma_2(\mathbb{F}_q^k, n, l, \varepsilon)$ is not aborted in step 4d, the statistical distance between $\mathcal{K}$'s view in $\Gamma_2(\mathbb{F}_q^k, n, l, \varepsilon)$ and $\mathcal{K}$'s view in $\Gamma_1(\mathbb{F}_q^k, n, l)$ is upper bounded by $n \cdot q^{-(2/3+\varepsilon)k}$. Finally, we just have to add the estimation from Lemma 59 to get the claimed upper bound for the statistical distance between $\mathcal{K}$'s view in $\Gamma_1(\mathbb{F}_q^k, n, l)$ and $\mathcal{K}$'s view in $\Gamma_2(\mathbb{F}_q^k, n, l, \varepsilon)$ without any conditions. □

Corollary 61. *The statistical distance between $\mathcal{K}$'s view in the game $\Gamma_2(\mathbb{F}_q^k, n, l, \varepsilon)$ and $\mathcal{K}$'s view in the game $\Gamma_3(\mathbb{F}_q^k, n, l, \varepsilon)$ is upper bounded by:*

$$n \cdot (q^{1-(1/3-\varepsilon)k} + q^{1-k} + q^{2-k}) + \sqrt{\exp(n \cdot q^{2-k}) - 1}$$

Proof. The only difference between $\Gamma_2(\mathbb{F}_q^k, n, l, \varepsilon)$ and $\Gamma_2(\mathbb{F}_q^k, n, l, \varepsilon)$ is that the abort message $(\star, i)$ in step 4d of $\Gamma_2(\mathbb{F}_q^k, n, l, \varepsilon)$ was replaced by $(\perp, i)$. Thus, the statistical distance between $\mathcal{K}$'s respective views is just the probability that the game $\Gamma_2(\mathbb{F}_q^k, n, l, \varepsilon)$ is aborted in step 4d. We already estimated this abort probability by the claimed term in Lemma 59. □

Lemma 62. *The statistical distance between $\mathcal{K}$'s view in the game $\Gamma_3(\mathbb{F}_q^k, n, l, \varepsilon)$ and $\mathcal{K}$'s view in the game $\Gamma_4(\mathbb{F}_q^k, n, l, \varepsilon)$ is upper bounded by $n \cdot q^{1-k/3}$, if $q^k \geq 2^{1/\varepsilon}$.*

Game $\Gamma_4(\mathbb{F}_q^k, n, l, \varepsilon)$

Parametrized by a finite vector space $\mathbb{F}_q^k$, some $n, l \in \mathbb{N}_{>0}$ with $l \leq n$, and $\varepsilon \in \mathbb{R}_{>0}$ such that $q^{(2/3+\varepsilon)k} \geq q$. Let $\mathcal{U} := \mathbb{F}_q^{1\times k}$ and $\mathcal{H} := \mathbb{F}_q^k \setminus \{0\}$. The player $\mathcal{K}$ is computationally unbounded.

1. For each $i \in \{1, \ldots, n\}$ the player $\mathcal{K}$ specifies some mapping $\tau_i : \mathcal{U}^i \to \mathbb{F}_q^{4k\times k}$.
2. The player $\mathcal{K}$ learns n random vectors $h_1, \ldots, h_n \xleftarrow{\text{r}} \mathcal{H}$ and a random matrix $C \xleftarrow{\text{r}} \mathbb{F}_q^{3k\times 4k}$.
3. A challenge bit $d \xleftarrow{\text{r}} \{0, 1\}$ is chosen secretly.
4. For $i = 1, \ldots, n$:
   a) The player $\mathcal{K}$ chooses some $\tilde{r}_i \in \mathbb{F}_q^{3k}$, $\tilde{S}_i \in \mathbb{F}_q^{3k\times k}$, $x_i \in \mathbb{F}_q$.
   b) If $i \leq l - d$, let $w_i := z_i \xleftarrow{\text{r}} \{\tilde{z} \in \mathcal{U} \mid \tilde{z}h_i = x_i\}$, else let $w_i := u_i \xleftarrow{\text{r}} \mathcal{U}$, chosen secretly.
   c) If there exists a unique tuple $(r_i, S_i) \in \mathbb{F}_q^{4k} \times \mathbb{F}_q^{4k\times k}$ such that $\tau_i(w_1, \ldots, w_i) = r_i w_i + S_i$ and $\#\{\tilde{v} \in \mathcal{U} \mid \tau_i(w_1, \ldots, w_{i-1}, \tilde{v}) = r_i\tilde{v} + S_i\} > q^{(2/3+\varepsilon)k}$, and for this unique tuple it holds that $(Cr_i, CS_i) = (\tilde{r}_i, \tilde{S}_i)$, then $\mathcal{K}$ learns $r_i x_i + S_i h_i$.
   Otherwise, $\mathcal{K}$ receives a special message $(\perp, i)$ and the game is aborted in the sense that step 5 follows next.
5. The player $\mathcal{K}$ computes and outputs a guess bit $\tilde{d} \in \{0, 1\}$. He wins the game, if $\tilde{d} = d$.

Figure 8.9: Fourth transformation of our stand-alone indistinguishability game. The only difference to $\Gamma_3(\mathbb{F}_q^k, n, l, \varepsilon)$ is the way the tuple (r_i, S_i) is computed in step 4c.

Proof. The only difference between $\Gamma_4(\mathbb{F}_q^k, n, l, \varepsilon)$ and $\Gamma_3(\mathbb{F}_q^k, n, l, \varepsilon)$ is in the computation of the tuple (r_i, S_i). It is straightforward to verify (see also Lemma 49) that by construction in step 4c of $\Gamma_3(\mathbb{F}_q^k, n, l, \varepsilon)$ it always holds: Either the game is aborted, or $\tau_i(w_1, \ldots, w_i) = r_i w_i + S_i$ and $\#\{\tilde{v} \in \mathcal{U} \mid \tau_i(w_1, \ldots, w_{i-1}, \tilde{v}) = r_i\tilde{v} + S_i\} > q^{(2/3+\varepsilon)k}$ and $(Cr_i, CS_i) = (\tilde{r}_i, \tilde{S}_i)$. Thus, we just have to estimate the probability that there exists some other tuple $(r', S') \in \mathbb{F}_q^{4k} \times \mathbb{F}_q^{4k\times k} \setminus \{(r_i, S_i)\}$ with $\tau_i(w_1, \ldots, w_i) = r'w_i + S'$ and $\#\{\tilde{v} \in \mathcal{U} \mid \tau_i(w_1, \ldots, w_{i-1}, \tilde{v}) = r'\tilde{v} + S'\} > q^{(2/3+\varepsilon)k}$. Now given that $q^k \geq 2^{1/\varepsilon}$, we have by Lemma 51 that only for less than $q^{2k/3}$ different choices of w_i there may exist such a second tuple (r', S'). For each stage $i \in \{1, \ldots, n\}$, since w_i is chosen uniformly random with support size q^{k-1} or larger, we can hence upper bound the probability that such a second tuple (r', S') exists by $q^{2k/3}/q^{k-1}$. Thus, our lemma follows by the Union bound. □

Lemma 63. *The maximum winning probability in the game* $\Gamma_4(\mathbb{F}_q^k, n, l, \varepsilon)$ *is upper bounded by the maximum winning probability in the game* $\Gamma_5(\mathbb{F}_q^k, n, l)$.

Proof. This holds trivially, since the player in $\Gamma_5(\mathbb{F}_q^k, n, l)$ is strictly stronger than in $\Gamma_4(\mathbb{F}_q^k, n, l, \varepsilon)$. □

Lemma 64. *The maximum winning probability in the game* $\Gamma_5(\mathbb{F}_q^k, n, l)$ *is upper bounded by the maximum winning probability in the game* $\Gamma_6(\mathbb{F}_q^k, n, l)$.

Proof. This holds trivially, since the player in $\Gamma_6(\mathbb{F}_q^k, n, l)$ is strictly stronger than in $\Gamma_5(\mathbb{F}_q^k, n, l)$. □

Lemma 65. *The games* $\Gamma_6(\mathbb{F}_q^k, n, l)$ *and* $\Gamma_7(\mathbb{F}_q^k, n, l)$ *have the same maximum winning probability.*

Proof. This holds trivially, since the changes from $\Gamma_6(\mathbb{F}_q^k, n, l)$ to $\Gamma_7(\mathbb{F}_q^k, n, l)$ are just cosmetic. □

Game $\Gamma_5(\mathbb{F}_q^k, n, l)$

Parametrized by a finite vector space $\mathbb{F}_q^k$ and some $n, l \in \mathbb{N}_{>0}$ with $l \leq n$. Let $\mathcal{U} := \mathbb{F}_q^{1\times k}$ and $\mathcal{H} := \mathbb{F}_q^k \setminus \{0\}$. The player $\mathcal{K}$ is computationally unbounded.

1. For each $i \in \{1, \ldots, n\}$ the player $\mathcal{K}$ specifies some mapping $\tilde{\tau}_i : \mathcal{U}^i \to \mathbb{F}_q^{4k\times(1+k)} \cup \{\perp\}$.
2. The player $\mathcal{K}$ learns n random vectors $h_1, \ldots, h_n \xleftarrow{r} \mathcal{H}$ and a random matrix $C \xleftarrow{r} \mathbb{F}_q^{3k\times 4k}$.
3. A challenge bit $d \xleftarrow{r} \{0, 1\}$ is chosen secretly.
4. For $i = 1, \ldots, n$:
   a) The player $\mathcal{K}$ chooses some $\tilde{r}_i \in \mathbb{F}_q^{3k}$, $\tilde{S}_i \in \mathbb{F}_q^{3k\times k}$, $x_i \in \mathbb{F}_q$.
   b) If $i \leq l - d$, let $w_i := z_i \xleftarrow{r} \{\tilde{z} \in \mathcal{U} \mid \tilde{z}h_i = x_i\}$, else let $w_i := u_i \xleftarrow{r} \mathcal{U}$, chosen secretly.
   c) If $C \cdot \tilde{\tau}_i(w_1, \ldots, w_i) = (\tilde{r}_i, \tilde{S}_i)$, then $\mathcal{K}$ learns $(r_i, S_i) := \tilde{\tau}_i(w_1, \ldots, w_i)$.
   Otherwise, $\mathcal{K}$ receives a special message $(\perp, i)$ and the game is aborted in the sense that step 5 follows next.
5. The player $\mathcal{K}$ computes and outputs a guess bit $\tilde{d} \in \{0, 1\}$. He wins the game, if $\tilde{d} = d$.

Figure 8.10: Fifth transformation of our stand-alone indistinguishability game. There are two differences to the game $\Gamma_4(\mathbb{F}_q^k, n, l, \varepsilon)$, which just make the player $\mathcal{K}$ strictly stronger. Firstly, the player $\mathcal{K}$ directly learns (r_i, S_i) instead of only $r_i x_i + S_i h_i$ in step 4c. Secondly, the tuple (r_i, S_i) in step 4c is no longer generated deterministically from τ_i and $w_1, \ldots, w_i$ by the game, but the player $\mathcal{K}$ may specify an arbitrary mapping $\tilde{\tau}_i$ instead that directly generates (r_i, S_i) from $w_1, \ldots, w_i$.

Lemma 66. *The maximum winning probability in the game $\Gamma_7(\mathbb{F}_q^k, n, l)$ and the maximum winning probability in the game $\Gamma_8(\mathbb{F}_q^k, n, l)$ differ at most by:*

$$n \cdot q^{1-k} + \sqrt{\exp(n \cdot q^{2-k}) - 1}$$

Proof. Let some arbitrary player $\mathcal{K}$ be given and let the random variables $\mathbf{C}, \mathbf{w}_1, \ldots, \mathbf{w}_l$ denote the same-named values in the game $\Gamma_7(\mathbb{F}_q^k, n, l)$. First of all, we just arbitrarily fix the random coins of $\mathcal{K}$ and hence get some fixed mappings $\tilde{\tau}_l, \ldots, \tilde{\tau}_n : \mathcal{U}^l \to \mathbb{F}_q^{4k\times(1+k)} \cup \{\perp\}$ in step 1 of $\Gamma_7(\mathbb{F}_q^k, n, l)$. Now note that, if $\mathbf{C}M \neq \mathbf{C}M'$ for all distinct $M, M' \in \tilde{\tau}_i(\mathbf{w}_1, \ldots, \mathbf{w}_{l-1}, \mathcal{U})$, then $\tilde{\tau}_i(\mathbf{w}_1, \ldots, \mathbf{w}_l)$ is completely determined by $(\mathbf{C}, \mathbf{w}_1, \ldots, \mathbf{w}_{l-1}, \mathbf{C} \cdot \tilde{\tau}_i(\mathbf{w}_1, \ldots, \mathbf{w}_l))$ and the specification of $\tilde{\tau}_i$. Thus, conditioned to the event that $\mathbf{C}M \neq \mathbf{C}M'$ for all distinct $M, M' \in \tilde{\tau}_i(\mathbf{w}_1, \ldots, \mathbf{w}_{l-1}, \mathcal{U})$ for all $i \in \{l, \ldots, n\}$ in both games $\Gamma_7(\mathbb{F}_q^k, n, l)$ and $\Gamma_8(\mathbb{F}_q^k, n, l)$, we can straightforwardly transform a player for $\Gamma_7(\mathbb{F}_q^k, n, l)$ into a player for $\Gamma_8(\mathbb{F}_q^k, n, l)$ with exactly the same winning probability. In other words, the maximum winning probability in the game $\Gamma_7(\mathbb{F}_q^k, n, l)$ may differ from the maximum winning probability in the game $\Gamma_8(\mathbb{F}_q^k, n, l)$ at most by the probability that $\mathbf{C}M = \mathbf{C}M'$ for some distinct $M, M' \in \tilde{\tau}_i(\mathbf{w}_1, \ldots, \mathbf{w}_{l-1}, \mathcal{U})$ with $i \in \{l, \ldots, n\}$. However, by Lemma 50 and the Union Bound we can estimate this probability by $(n - l + 1) \cdot q^{1-3k} \cdot |\mathcal{U}|^2 + \iota(\mathbf{C}, (\mathbf{w}_1, \ldots, \mathbf{w}_{l-1}))$. Further, by Corollary 48 we have that $\iota(\mathbf{C}, (\mathbf{w}_1, \ldots, \mathbf{w}_{l-1})) < \sqrt{\exp((l-1)q^{2-k}) - 1}$. Together this yields the claimed estimation. □

Lemma 67. *The games $\Gamma_8(\mathbb{F}_q^k, n, l)$ and $\Gamma_9(\mathbb{F}_q^k, n, l)$ have the same maximum winning probability.*

Proof. This holds trivially, since w.l.o.g. we only need to consider deterministic players. □

Lemma 68. *The maximum winning probability in the game $\Gamma_9(\mathbb{F}_q^k, n, l)$ is upper bounded by:*

$$\tfrac{1}{2} + n \cdot \left(2q^{(4-k)/3} + q \cdot \sqrt{\exp(n \cdot q^{2-k}) - 1}\right)$$

Game $\Gamma_6(\mathbb{F}_q^k, n, l)$

Parametrized by a finite vector space $\mathbb{F}_q^k$ and some $n, l \in \mathbb{N}_{>0}$ with $l \leq n$. Let $\mathcal{U} := \mathbb{F}_q^{1\times k}$ and $\mathcal{H} := \mathbb{F}_q^k \setminus \{0\}$. The player $\mathcal{K}$ is computationally unbounded.

1. a) For each $i \in \{l, \ldots, n\}$ the player $\mathcal{K}$ specifies some mapping $\tilde{\tau}_i : \mathcal{U}^i \to \mathbb{F}_q^{4k\times(1+k)} \cup \{\perp\}$.
   b) For each $i \in \{l+1, \ldots, n\}$ the player $\mathcal{K}$ chooses some $u_i \in \mathcal{U}$.
2. The player $\mathcal{K}$ learns n random vectors $h_1, \ldots, h_n \xleftarrow{r} \mathcal{H}$ and a random matrix $C \xleftarrow{r} \mathbb{F}_q^{3k\times 4k}$.
3. For $i = 1, \ldots, l-1$: The player $\mathcal{K}$ chooses some $x_i \in \mathbb{F}_q$ and learns $w_i := z_i \xleftarrow{r} \{\tilde{z} \in \mathcal{U} \mid \tilde{z}h_i = x_i\}$.
4. a) A challenge bit $d \xleftarrow{r} \{0,1\}$ is chosen secretly, and $\mathcal{K}$ chooses some $\tilde{r}_l \in \mathbb{F}_q^{3k}$, $\tilde{S}_l \in \mathbb{F}_q^{3k\times k}$, $x_l \in \mathbb{F}_q$.
   b) If $d = 0$, let $w_l := z_l \xleftarrow{r} \{\tilde{z} \in \mathcal{U} \mid \tilde{z}h_l = x_l\}$, else let $w_l := u_l \xleftarrow{r} \mathcal{U}$, chosen secretly.
   c) If $C \cdot \tilde{\tau}_l(w_1, \ldots, w_l) = (\tilde{r}_l, \tilde{S}_l)$, then $\mathcal{K}$ learns $(r_l, S_l) := \tilde{\tau}_l(w_1, \ldots, w_l)$.
   Otherwise, $\mathcal{K}$ receives a special message $(\perp, l)$ and the game is aborted in the sense that step 6 follows next.
5. For $i = l+1, \ldots, n$:
   a) The player $\mathcal{K}$ chooses some $\tilde{r}_i \in \mathbb{F}_q^{3k}$, $\tilde{S}_i \in \mathbb{F}_q^{3k\times k}$. Let $w_i := u_i$.
   b) If $C \cdot \tilde{\tau}_i(w_1, \ldots, w_i) = (\tilde{r}_i, \tilde{S}_i)$, then $\mathcal{K}$ learns $(r_i, S_i) := \tilde{\tau}_i(w_1, \ldots, w_i)$.
   Otherwise, $\mathcal{K}$ receives a special message $(\perp, i)$ and the game is aborted in the sense that step 6 follows next.
6. The player $\mathcal{K}$ computes and outputs a guess bit $\tilde{d} \in \{0,1\}$. He wins the game, if $\tilde{d} = d$.

Figure 8.11: Sixth transformation of our stand-alone indistinguishability game. There are two differences to the game $\Gamma_5(\mathbb{F}_q^k, n, l)$, which just make the player $\mathcal{K}$ strictly stronger. Firstly, the last $n - l$ "token inputs" $w_{l+1}, \ldots, w_n$ are no longer chosen uniformly at random, but the player $\mathcal{K}$ may choose them at the start of the game in step 1b. Secondly, in the first $l - 1$ stages the game may no longer be aborted, and the player $\mathcal{K}$ directly learns w_i instead of only $\tilde{\tau}_i(w_1, \ldots, w_i)$, which makes the mappings $\tilde{\tau}_1, \ldots, \tilde{\tau}_{l-1}$ obsolete.

Proof. W.l.o.g. we consider a deterministic player $\mathcal{K}$, i.e. the mappings $\tilde{\tau}_i, x_i, \tilde{\sigma}_i$ are all fixed. Let the random variables $\mathbf{h}_1, \ldots, \mathbf{h}_n, \mathbf{z}_1, \ldots, \mathbf{z}_{l-1}, \mathbf{w}, \mathbf{d}$ represent the same-named random values in the game $\Gamma_9(\mathbb{F}_q^k, n)$, i.e. it holds:

$$\mathbf{d} \xleftarrow{r} \{0,1\} \qquad \mathbf{h}_1, \ldots, \mathbf{h}_l \xleftarrow{r} \mathcal{H} \qquad \mathbf{z}_i \xleftarrow{r} \{z \in \mathcal{U} \mid z\mathbf{h}_i = x_i(\mathbf{h}_1, \ldots, \mathbf{h}_l, \mathbf{z}_1, \ldots, \mathbf{z}_{i-1})\}$$

For convenience we set:

$$\mathbf{H} := (\mathbf{h}_1, \ldots, \mathbf{h}_l) \qquad \mathbf{H}' := (\mathbf{h}_1, \ldots, \mathbf{h}_{l-1}) \qquad \mathbf{T} := (\mathbf{z}_1, \ldots, \mathbf{z}_{l-1}, \mathbf{w}) \qquad \mathbf{T}' := (\mathbf{z}_1, \ldots, \mathbf{z}_{l-1})$$

Further, let the random variable $\mathbf{m} \in \{l-1, \ldots, n\}$ represent the index of the latest stage where the game is not aborted; i.e. $\tilde{\tau}_i(\mathbf{T}) = \tilde{\sigma}_i(\mathbf{H}, \mathbf{T}')$ for all $i \in \{l, \ldots, \mathbf{m}\}$, and $\tilde{\tau}_{\mathbf{m}+1}(\mathbf{T}) \neq \tilde{\sigma}_{\mathbf{m}+1}(\mathbf{H}, \mathbf{T}')$ if not $\mathbf{m} = n$. Note that $\mathcal{K}$'s complete view can be deterministically reconstructed from $(\mathbf{H}, \mathbf{T}', \mathbf{m})$ and $\mathcal{K}$'s program code. Thus, with the random variable $\tilde{\mathbf{d}}$ representing $\mathcal{K}$'s final guess, we have:

$$\begin{aligned}
&\mathbb{P}[\tilde{\mathbf{d}} = \mathbf{d}] \qquad (8.1)\\
&= \mathbb{P}[\tilde{\mathbf{d}} = 0 \,|\, \mathbf{d} = 0] \cdot \mathbb{P}[\mathbf{d} = 0] + \mathbb{P}[\tilde{\mathbf{d}} = 1 \,|\, \mathbf{d} = 1] \cdot \mathbb{P}[\mathbf{d} = 1]\\
&= \tfrac{1}{2}\Big(\mathbb{P}[\tilde{\mathbf{d}} = 0 \,|\, \mathbf{d} = 0] + \mathbb{P}[\tilde{\mathbf{d}} = 1 \,|\, \mathbf{d} = 1]\Big)\\
&= \tfrac{1}{2}\Big(\mathbb{P}[\tilde{\mathbf{d}} = 0 \,|\, \mathbf{d} = 0] + 1 - \mathbb{P}[\tilde{\mathbf{d}} = 0 \,|\, \mathbf{d} = 1]\Big)\\
&\leq \tfrac{1}{2} + \tfrac{1}{2}\Big|\mathbb{P}[\tilde{\mathbf{d}} = 0 \,|\, \mathbf{d} = 0] - \mathbb{P}[\tilde{\mathbf{d}} = 0 \,|\, \mathbf{d} = 1]\Big|\\
&\leq \tfrac{1}{2} + \tfrac{1}{2}\sum\nolimits_{H,T',m}\Big|\mathbb{P}[(\mathbf{H}, \mathbf{T}', \mathbf{m}) = (H, T', m) \,|\, \mathbf{d} = 0] - \mathbb{P}[(\mathbf{H}, \mathbf{T}', \mathbf{m}) = (H, T', m) \,|\, \mathbf{d} = 1]\Big|
\end{aligned}$$

Game $\Gamma_7(\mathbb{F}_q^k, n, l)$

Parametrized by a finite vector space $\mathbb{F}_q^k$ and some $n, l \in \mathbb{N}_{>0}$ with $l \leq n$. Let $\mathcal{U} := \mathbb{F}_q^{1\times k}$ and $\mathcal{H} := \mathbb{F}_q^k \backslash \{0\}$. The player $\mathcal{K}$ is computationally unbounded.

1. For each $i \in \{l, \ldots, n\}$ the player $\mathcal{K}$ specifies some mapping $\tilde{\tau}_i : \mathcal{U}^l \to \mathbb{F}_q^{4k\times(1+k)} \cup \{\perp\}$.
2. The player $\mathcal{K}$ learns l random vectors $h_1, \ldots, h_l \xleftarrow{r} \mathcal{H}$ and a random matrix $C \xleftarrow{r} \mathbb{F}_q^{3k\times 4k}$.
3. For $i = 1, \ldots, l-1$: The player $\mathcal{K}$ chooses some $x_i \in \mathbb{F}_q$ and learns $w_i := z_i \xleftarrow{r} \{\tilde{z} \in \mathcal{U} \mid \tilde{z}h_i = x_i\}$.
4. a) A challenge bit $d \xleftarrow{r} \{0, 1\}$ is chosen secretly, and $\mathcal{K}$ chooses some $x_l \in \mathbb{F}_q$.
   b) If $d = 0$, let $w_l := z_l \xleftarrow{r} \{\tilde{z} \in \mathcal{U} \mid \tilde{z}h_l = x_l\}$, else let $w_l := u_l \xleftarrow{r} \mathcal{U}$, chosen secretly.
5. For $i = l, \ldots, n$:
   a) The player $\mathcal{K}$ chooses some $\tilde{r}_i \in \mathbb{F}_q^{3k}$, $\tilde{S}_i \in \mathbb{F}_q^{3k\times k}$.
   b) If $C \cdot \tilde{\tau}_i(w_1, \ldots, w_l) = (\tilde{r}_i, \tilde{S}_i)$, then $\mathcal{K}$ learns $(r_i, S_i) := \tilde{\tau}_i(w_1, \ldots, w_l)$.
   Otherwise, $\mathcal{K}$ receives a special message $(\perp, i)$ and the game is aborted in the sense that step 6 follows next.
6. The player $\mathcal{K}$ computes and outputs a guess bit $\tilde{d} \in \{0, 1\}$. He wins the game, if $\tilde{d} = d$.

Figure 8.12: Seventh transformation of our stand-alone indistinguishability game. This is just a "cleaned" version of $\Gamma_6(\mathbb{F}_q^k, n, l)$. Firstly, instead of letting the player $\mathcal{K}$ choose the last $n - l$ "token inpute" $w_{l+1}, \ldots, w_n$ at the start of the game explicitly, they are now implicitly hard-coded. Secondly, the meanwhile obsolete random vectors $h_{l+1}, \ldots, h_n$ are omitted. Thirdly, we moved $\mathcal{K}$'s choice of $(\tilde{r}_l, \tilde{S}_l)$ and the subsequent output generation from step 4 to step 5.

Now, for $H \in \mathcal{H}^l$, $T' \in \mathcal{U}^{l-1}$, $m \in \{l-1, \ldots, n\}$ we define the following sets:

$$A_m(H, T') := \{\tilde{v} \in \mathcal{U} \mid \forall\, j \in \{l, \ldots, m\} :\ \tilde{\tau}_j(T', \tilde{v}) = \tilde{\sigma}_j(H, T')\}$$
$$\bar{A}_m(H, T') := A_m \backslash A_{m+1} \quad \text{with the convention that } A_{n+1}(H, T') = \emptyset$$

The intuition behind this is that $A_m(\mathbf{H}, \mathbf{T}')$ consists of all token inputs for stage l, such that the game is not aborted before stage m. Accordingly, $\bar{A}_m(\mathbf{H}, \mathbf{T}')$ consists of all token inputs for stage l, such that stage m is the latest non-aborted stage. In other words, it holds:

$$A_m(H, T') = \{w \in \mathcal{U} \mid (\mathbf{H}, \mathbf{T}', \mathbf{w}) = (H, T', w) \ \Rightarrow\ \mathbf{m} \geq m\}$$
$$\bar{A}_m(H, T') = \{w \in \mathcal{U} \mid (\mathbf{H}, \mathbf{T}', \mathbf{w}) = (H, T', w) \ \Rightarrow\ \mathbf{m} = m\}$$

Further, for all $h \in \mathcal{H}$, $\alpha \in \mathbb{F}_q$ we define:

$$Z_\alpha(h) := \{\tilde{z} \in \mathcal{U} \mid zh = \alpha\}$$

Note that $\mathbf{w} \xleftarrow{r} Z_{x_l(\mathbf{H},\mathbf{T}')}(\mathbf{h}_l)$ if $\mathbf{d} = 0$, and $\mathbf{w} \xleftarrow{r} \mathcal{U}$ if $\mathbf{d} = 1$. Hence, given $H := (h_1, \ldots, h_l) \in \mathcal{H}^l$, $T' \in \mathcal{U}^{l-1}$, $m \in \{l-1, \ldots, n\}$, we can compute:

$$\begin{aligned}
&\Big|\mathbb{P}\big[\mathbf{m} = m \mid (\mathbf{d}, \mathbf{H}, \mathbf{T}') = (0, H, T')\big] - \mathbb{P}\big[\mathbf{m} = m \mid (\mathbf{d}, \mathbf{H}, \mathbf{T}') = (1, H, T')\big]\Big| \\
&= \Big|\mathbb{P}\big[\mathbf{w} \in \bar{A}_m(H, T') \mid (\mathbf{d}, \mathbf{H}, \mathbf{T}') = (0, H, T')\big] - \mathbb{P}\big[\mathbf{w} \in \bar{A}_m(H, T') \mid (\mathbf{d}, \mathbf{H}, \mathbf{T}') = (1, H, T')\big]\Big| \\
&= \left|\frac{\big|Z_{x_l(H,T')}(h_l) \cap \bar{A}_m(H, T')\big|}{\big|Z_{x_l(H,T')}(h_l)\big|} - \frac{|\bar{A}_m(H, T')|}{|\mathcal{U}|}\right| \\
&= q^{1-k} \cdot \Big|\big|Z_{x_l(H,T')}(h_l) \cap \bar{A}_m(H, T')\big| - \tfrac{1}{q}\big|\bar{A}_m(H, T')\big|\Big|
\end{aligned}$$

Game $\Gamma_8(\mathbb{F}_q^k, n, l)$

Parametrized by a finite vector space $\mathbb{F}_q^k$ and some $n, l \in \mathbb{N}_{>0}$ with $l \leq n$. Let $\mathcal{U} := \mathbb{F}_q^{1\times k}$ and $\mathcal{H} := \mathbb{F}_q^k \setminus \{0\}$. The player $\mathcal{K}$ is computationally unbounded.

1. For each $i \in \{l, \ldots, n\}$ the player $\mathcal{K}$ specifies some mapping $\tilde{\tau}_i : \mathcal{U}^l \to \mathbb{F}_q^{4k\times(1+k)} \cup \{\perp\}$.
2. The player $\mathcal{K}$ learns l random vectors $h_1, \ldots, h_l \xleftarrow{\text{r}} \mathcal{H}$ and a random matrix $C \xleftarrow{\text{r}} \mathbb{F}_q^{3k\times 4k}$.
3. For $i = 1, \ldots, l-1$: The player $\mathcal{K}$ chooses some $x_i \in \mathbb{F}_q$ and learns $w_i := z_i \xleftarrow{\text{r}} \{\tilde{z} \in \mathcal{U} \mid \tilde{z}h_i = x_i\}$.
4. a) A challenge bit $d \xleftarrow{\text{r}} \{0,1\}$ is chosen secretly, and $\mathcal{K}$ chooses some $x_l \in \mathbb{F}_q$.
   b) If $d = 0$, let $w_l := z_l \xleftarrow{\text{r}} \{\tilde{z} \in \mathcal{U} \mid \tilde{z}h_i = x_i\}$, else let $w_l := u_l \xleftarrow{\text{r}} \mathcal{U}$, chosen secretly.
5. For $i = l, \ldots, n$:
   a) The player $\mathcal{K}$ chooses some $r_i \in \mathbb{F}_q^{4k}$, $S_i \in \mathbb{F}_q^{4k\times k}$.
   b) If $\tilde{\tau}_i(w_1, \ldots, w_l) = (r_i, S_i)$, then $\mathcal{K}$ is notified about that by a special message $(\top, i)$. Otherwise, $\mathcal{K}$ receives a special message $(\perp, i)$ and the game is aborted in the sense that step 6 follows next.
6. The player $\mathcal{K}$ computes and outputs a guess bit $\tilde{d} \in \{0,1\}$. He wins the game, if $\tilde{d} = d$.

Figure 8.13: Eighth transformation of our stand-alone indistinguishability game. The only difference to $\Gamma_7(\mathbb{F}_q, n, l)$ is that in step 5b the player $\mathcal{K}$ now must exactly forecast $\tilde{\tau}_i(w_1, \ldots, w_l)$ rather than only the linear projection $C \cdot \tilde{\tau}_i(w_1, \ldots, w_l)$.

Plugging this into (8.1), we get:

$$\mathbb{P}[\tilde{\mathbf{d}} = \mathbf{d}] \;\leq\; \tfrac{1}{2} + \tfrac{q^{1-k}}{2} \sum_{m=l-1}^{n} \mathbb{E}\left|\left|Z_{x_l(\mathbf{H},\mathbf{T}')}(\mathbf{h}_l) \cap \bar{A}_m(\mathbf{H},\mathbf{T}')\right| - \tfrac{1}{q}\left|\bar{A}_m(\mathbf{H},\mathbf{T}')\right|\right|$$

Now we exploit that $\left|\bar{A}_m(H,T')\right| = \left|A_m(H,T') \setminus A_{m-1}(H,T')\right| = \left|A_m(H,T')\right| - \left|A_{m-1}(H,T')\right|$ by construction and analogously $\left|Z \cap \bar{A}_m(H,T')\right| = \left|Z \cap A_m(H,T')\right| - \left|Z \cap A_{m-1}(H,T')\right|$ for every $Z \subseteq \mathcal{U}$. Using this and the Triangle Inequality, we can derive:

$$\mathbb{P}[\tilde{\mathbf{d}} = \mathbf{d}] \;\leq\; \tfrac{1}{2} + q^{1-k} \sum_{m=l-1}^{n+1} \mathbb{E}\left|\left|Z_{x_l(\mathbf{H},\mathbf{T}')}(\mathbf{h}_l) \cap A_m(\mathbf{H},\mathbf{T}')\right| - \tfrac{1}{q}\left|A_m(\mathbf{H},\mathbf{T}')\right|\right|$$

Since always $A_{l-1}(\mathbf{H},\mathbf{T}') = \mathbb{F}_q^{1\times k}$ and $A_{n+1}(\mathbf{H},\mathbf{T}') = \emptyset$ by definition, the first and last summand of this expression are always zero and can be discarded; i.e. it holds:

$$\mathbb{P}[\tilde{\mathbf{d}} = \mathbf{d}] \;\leq\; \tfrac{1}{2} + q^{1-k} \sum_{m=l}^{n} \mathbb{E}\left|\left|Z_{x_l(\mathbf{H},\mathbf{T}')}(\mathbf{h}_l) \cap A_m(\mathbf{H},\mathbf{T}')\right| - \tfrac{1}{q}\left|A_m(\mathbf{H},\mathbf{T}')\right|\right| \tag{8.2}$$

Now we exploit that $\left\{A_m(H,T')\right\}_{H\in\mathcal{H}^l}$ can be considered as a disjoint decomposition of some subset of $\mathbb{F}_q^{1\times k}$, since by construction we have:

$$A_m(H_1,T') \neq A_m(H_2,T') \quad\Rightarrow\quad A_m(H_1,T') \cap A_m(H_2,T') = \emptyset$$

Thus, for arbitrary $\gamma \in \mathbb{R}_{>0}$ by Corollary 54 follows:

$$\mathbb{P}\left[\exists\, \alpha \in \mathbb{F}_q,\, H \in \mathcal{H}^l : \left|\left|Z_\alpha(\mathbf{h}_l) \cap A_m(H,\mathbf{T}')\right| - \tfrac{1}{q}\left|A_m(H,\mathbf{T}')\right|\right| > \gamma\right] \;\leq\; \frac{q^{k+1/2}}{\gamma^{3/2}} + \iota(\mathbf{h}_l,\mathbf{T}')$$

We instantiate α in this inequality by $x_l(\mathbf{H},\mathbf{T}')$ and H by $\mathbf{H}$, which yields:

$$\mathbb{P}\left[\left|\left|Z_{x_l(\mathbf{H},\mathbf{T}')}(\mathbf{h}_l) \cap A_m(\mathbf{H},\mathbf{T}')\right| - \tfrac{1}{q}\left|A_m(\mathbf{H},\mathbf{T}')\right|\right| > \gamma\right] \;\leq\; \frac{q^{k+1/2}}{\gamma^{3/2}} + \iota(\mathbf{h}_l,\mathbf{T}')$$

Game $\Gamma_9(\mathbb{F}_q^k, n, l)$

Parametrized by a finite vector space $\mathbb{F}_q^k$ and some $n, l \in \mathbb{N}_{>0}$ with $l \leq n$. Let $\mathcal{U} := \mathbb{F}_q^{1\times k}$ and $\mathcal{H} := \mathbb{F}_q^k \setminus \{0\}$. The player $\mathcal{K}$ is computationally unbounded.

1. a) For each $i \in \{1, \ldots, l\}$ the player $\mathcal{K}$ specifies some mapping $x_i : \mathcal{H}^l \times \mathcal{U}^{i-1} \to \mathbb{F}_q$.
   b) For each $i \in \{l, \ldots, n\}$ the player $\mathcal{K}$ specifies some mapping $\tilde{\tau}_i : \mathcal{U}^l \to \mathbb{F}_q^{4k\times(1+k)} \cup \{\perp\}$.
   c) For each $i \in \{l, \ldots, n\}$ the player $\mathcal{K}$ specifies some mapping $\tilde{\sigma}_i : \mathcal{H}^l \times \mathcal{U}^{l-1} \to \mathbb{F}_q^{4k\times(1+k)}$.
2. The player $\mathcal{K}$ learns l random vectors $h_1, \ldots, h_l \xleftarrow{\text{r}} \mathcal{H}$.
3. For $i = 1, \ldots, l-1$: The player $\mathcal{K}$ learns $z_i \xleftarrow{\text{r}} \{\tilde{z} \in \mathcal{U} \mid \tilde{z}h_i = x_i(h_1, \ldots, h_l, z_1, \ldots, z_{i-1})\}$.
4. a) A challenge bit $d \xleftarrow{\text{r}} \{0, 1\}$ is chosen secretly.
   b) If $d = 0$, let $w \xleftarrow{\text{r}} \{\tilde{z} \in \mathcal{U} \mid \tilde{z}h_l = x_l(h_1, \ldots, h_n, z_1, \ldots, z_{l-1})\}$, else let $w \xleftarrow{\text{r}} \mathcal{U}$, chosen secretly.
5. For $i = l, \ldots, n$: If $\tilde{\tau}_i(z_1, \ldots, z_{l-1}, w) = \tilde{\sigma}_i(h_1, \ldots, h_l, z_1, \ldots, z_{l-1})$, then $\mathcal{K}$ is notified about that by a special message $(\top, i)$; else $\mathcal{K}$ receives a special message $(\perp, i)$ and the game is aborted in the sense that step 6 follows next.
6. The player $\mathcal{K}$ computes and outputs a guess bit $\tilde{d} \in \{0, 1\}$. He wins the game, if $\tilde{d} = d$.

Figure 8.14: Final transformation of our stand-alone indistinguishability game. The difference to the game $\Gamma_8(\mathbb{F}_q^k, n, l)$ is that the player $\mathcal{K}$ must specify in step 1 how all his future choices will depend on the information gathered so far, and the meanwhile obsolete random matrix C is omitted.

Since $\mathbb{E}(\mathbf{x}) = \int_0^\infty \mathbb{P}[\mathbf{x} > \gamma]\,\mathrm{d}\gamma$ for every real-valued random variable $\mathbf{x} \in \mathbb{R}_{\geq 0}$, this directly implies:

$$\begin{aligned}
&\mathbb{E}\left|\left|Z_{x_l(\mathbf{H},\mathbf{T}')}(\mathbf{h}_l) \cap A_m(\mathbf{H}, \mathbf{T}')\right| - \tfrac{1}{q}\left|A_m(\mathbf{H}, \mathbf{T}')\right|\right| \;\leq\; \int_0^{q^k} \min\{1, \tfrac{q^{k+1/2}}{\gamma^{3/2}}\} + \iota(\mathbf{h}_l, \mathbf{T}')\;\mathrm{d}\gamma \\
&= \; q^{(2k+1)/3} + \int_{q^{(2k+1)/3}}^{q^k} \tfrac{q^{k+1/2}}{\gamma^{3/2}}\;\mathrm{d}\gamma + q^k \cdot \iota(\mathbf{h}_l, \mathbf{T}') \;=\; 2q^{(2k+1)/3} - q^{(k+1)/2} + q^k \cdot \iota(\mathbf{h}_l, \mathbf{T}')
\end{aligned}$$

Moreover, by Corollary 48 we have that $\iota(\mathbf{h}_l, \mathbf{T}') < \sqrt{\exp\big((l-1)q^{2-k}\big) - 1}$. Using (8.2), we conclude:

$$\begin{aligned}
\mathbb{P}[\tilde{\mathbf{d}} = \mathbf{d}] \;&<\; \tfrac{1}{2} + q^{1-k} \cdot (n - l + 1) \cdot \left(2q^{(2k+1)/3} - q^{(k+1)/2} + q^k \cdot \sqrt{\exp\big((l-1)\cdot q^{2-k}\big) - 1}\right) \\
&<\; \tfrac{1}{2} + n \cdot \left(2q^{(4-k)/3} + q \cdot \sqrt{\exp(n \cdot q^{2-k}) - 1}\right) \qquad \square
\end{aligned}$$

8.3.8 Concluding the security proof

We can now finally conclude our security proof by just putting things together. We first sum up what we know so far about successive hybrid games; then we conclude this whole section with our final security theorem.

Corollary 69. *For any $l \in \{1, \ldots, n\}$, the hybrid games Game_{l-1} and Game_l are statistically indistinguishable, if $k \geq 5$. More particular, the statistical distance between the environment's respective views is negligible in the security parameter $\lambda := k \log q$, if only $k \geq 5$.*

Proof. For $i = 0, \ldots, 9$, let δ_i denote the player's advantage in the respective indistinguishability game; i.e. the maximum winning probability in the game $\Gamma_i(\mathbb{F}_q^k, n, l)$, or $\Gamma_i(\mathbb{F}_q^k, n, l, \varepsilon)$ respectively, is $\frac{1}{2} + \delta_i$. By Lemma 57, the statistical distance between the environment's views in Game_{l-1} and Game_l is upper bounded by $2\delta_0$. Furthermore, given any $\varepsilon \in \mathbb{R}_{>0}$ with $q^{(2/3+\varepsilon)k} \geq q$, it holds:

$$\begin{aligned}
&\delta_0 \leq \delta_1 && \text{by Lemma 58}\\
&\delta_1 \leq \delta_2 + n \cdot \left(q^{-(2/3+\varepsilon)k} + q^{1-(1/3-\varepsilon)k} + q^{1-k} + q^{2-k}\right) + \sqrt{\exp\left(n \cdot q^{2-k}\right) - 1} && \text{by Lemma 60}\\
&\delta_2 \leq \delta_3 + n \cdot \left(q^{1-(1/3-\varepsilon)k} + q^{1-k} + q^{2-k}\right) + \sqrt{\exp\left(n \cdot q^{2-k}\right) - 1} && \text{by Corollary 61}\\
&\delta_3 \leq \delta_4 + n \cdot q^{1-k/3}, \quad \text{if } q^k \geq 2^{1/\varepsilon} && \text{by Lemma 62}\\
&\delta_4 \leq \delta_5 && \text{by Lemma 63}\\
&\delta_5 \leq \delta_6 && \text{by Lemma 64}\\
&\delta_6 = \delta_7 && \text{by Lemma 65}\\
&\delta_7 \leq \delta_8 + n \cdot q^{1-k} + \sqrt{\exp\left(n \cdot q^{2-k}\right) - 1} && \text{by Lemma 66}\\
&\delta_8 = \delta_9 && \text{by Lemma 67}\\
&\delta_9 \leq n \cdot \left(2q^{(4-k)/3} + q \cdot \sqrt{\exp\left(n \cdot q^{2-k}\right) - 1}\right) && \text{by Lemma 68}
\end{aligned}$$

Now, let $\varepsilon := \frac{1}{12}$ and let $k \geq 5$, which especially yields that $q^{(2/3+\varepsilon)k} \geq q$ and allows us to estimate:

$$q^{-(2/3+\varepsilon)k},\ q^{1-(1/3-\varepsilon)k},\ q^{1-k},\ q^{2-k},\ q^{1-(1/3-\varepsilon)k},\ q^{1-k/3},\ q^{(4-k)/3} \ \leq\ q^{-k/5}$$

Further let $q^k \geq n^{25/3}$. This, together with $k \geq 5$, allows us to estimate:

$$q \cdot \sqrt{\exp\left(n \cdot q^{2-k}\right) - 1} \ \leq\ q \cdot \sqrt{\exp\left(q^{2-22k/25}\right) - 1} \ <\ q \cdot \sqrt{4q^{2-22k/25}} \ =\ 2q^{2-11k/25} \ \leq\ 2q^{-k/5}$$

Putting things together, we have shown that the statistical distance between the environment's views in the hybrid games Game_{l-1} and Game_l is upper bounded by $(13n + 3) \cdot \exp(-\lambda/5)$, where $\lambda := k \log q$ is the security parameter and we need that $\exp(\lambda) \geq \max\left(2^{12}, n^{25/3}\right)$. □

Theorem 70. *Let some arbitrary environment $\mathcal{Z}$ be given and some adversary $\mathcal{A}$ that corrupts the sender Goliath. Then the view of $\mathcal{Z}$ in the ideal model with ideal functionality $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$ and simulator $\mathcal{S}^{\mathrm{Goliath}}(\mathcal{A})$ is statistically indistinguishable (with security parameter $\lambda := k \log q$) from the view of $\mathcal{Z}$ in the real model with protocol $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$ and adversary $\mathcal{A}$, if only $k \geq 5$.*

Proof. By Corollary 69 we have that the statistical distance between the environment's views in successive hybrid games $\mathsf{Game}_{l-1}, \mathsf{Game}_l$ is negligible in the security parameter λ, if only $k \geq 5$. By the Union Bound, we can conclude that the statistical distance between the environment's views in Game_0 and Game_n may be at most by a factor n bigger, and hence is still negligible. Finally, by Corollary 56 we have that Game_0 is statistical indistinguishable from the ideal model, and Game_n is perfectly indistinguishable from the real model. Thus, the ideal model and the real model must be statistical indistinguishable. □

9 No-go arguments & conclusion

In this section we conclude our work by a short summery of what we achieved so far, what further improvement opportunities are left open and which drawbacks of our work seem unavoidable (or at least hard to circumvent). We start with the negative aspects; they highlight that our results are quite close to optimal. Though, we only give rather intuitive arguments than full formal proofs.

9.1 Impossibility of polynomially bounded simulation runtime

The running time of our simulator $\mathcal{S}^{\text{Goliath}}(\mathcal{A})$ for a corrupted sender is not a priori polynomially bounded (cf. Section 8.3). Instead, we have only a polynomial bound for the *expected* running time (cf. Lemma 55). The same problem occurred in [MS08] and they stated it as an open problem to find a protocol with strict polynomial-time simulation. We argue that such a protocol seems very hard to find, unless computational assumptions are used.

Since information-theoretically secure OT cannot be realized from stateless tokens, as shown by [GIMS10], it suffices to consider stateful solutions. However, simulatability is only possible if a corrupted sender's inputs can be extracted from his messages sent to the receiver party and the program code of the token(s). The straightforward approach of extraction is to rewind the token, but as the token may act honestly only on some fraction of inputs, the simulator will have to rewind the token repeatedly. In particular, a corrupted token issuer can choose some arbitrary probability p, such that the token acts honestly only with this probability p. Unless p is negligible, this will necessitate a simulator that can rewind the token for about $\frac{1}{p}$ times. Since p may be effectively chosen by the adversary (and thus by the environment) during runtime, strict polynomial-time simulation with repeated token rewinding seems impossible. Moreover, we are not aware of any information-theoretic approach (i.e. without computational assumptions) that would allow us to avoid repeated token rewinding.

9.2 Impossibility of random access solutions with a constant number of tokens

Via our protocol $\Pi_{\text{OAFE}}^{\text{semi-int}}$ one can implement sequentially queriable OTM tokens from a single piece of untrusted tamper-proof hardware (cf. Section 7.1 and Section 6.4). We discuss now, why it seems impossible to implement multiple OTMs that the token receiver can access in arbitrary order. The main argument is that a corrupted token issuer can try to let the token work only for the first OTM query and then shut down. This is not simulatable in the ideal model, since the simulator does not learn which OTM is queried first—the decision which OTM to query first even might be made not until the interactive part of the protocol is over.

In particular, the attack idea is as follows. Given any hypothetical protocol for random access OTMs from a single token, let b denote a lower bound of token queries that are needed for the first OTM access and let B denote an upper bound. W.l.o.g., b and B are polynomially bounded in the security parameter. The corrupted token issuer randomly picks $j \xleftarrow{r} \{b, \ldots, B\}$ and programs the token such that it shuts down after the j-th query. Now, with probability $\frac{1}{B-b+1}$ the receiver party will be able to access only the very OTM that is queried first. Note that this probability is independent of the access order to the implemented OTMs. Further note that by this attack it cannot happen that the OTM accessed first is malformed and any other is not. For the simulator

this means an unsolvable dilemma. With non-negligible probability, all but one of the sent OTMs must be malformed and the non-malformed OTM must always be that one that will be accessed first.

9.3 Lower bounds for David's communication overhead

Even our refined construction for l-bit string-OT (q.v. Section 7.2.1) needs that David inputs $\Theta(l)$ bits into the token. One could wonder, if it is possible to implement multiple instances of OT from reusable tamper-proof tokens, such that for each implemented instance of OT the communication complexity for the receiver party David is constant. We argue that this seems very improbable. The main argument is that a corrupted sender Goliath can correctly guess David's token inputs for the first OT instances with some constant probability. Thus, he can maliciously create the tokens so that they immediately shut down, if David's first token inputs do not match Goliath's guess. Thereby, when Goliath learns that the protocol was *not* aborted, he can reconstruct David's first OT input. Such a protocol cannot be UC-secure, since in the ideal model the abort probability may not depend on Davids inputs. Moreover, the whole argumentation still seems valid, even if we allow that David inputs polylogarithmically many bits per OT into the tokens.

9.4 Conclusion & improvement opportunities

We showed that a single untrusted tamper-proof hardware token is sufficient for non-interactive (or to be more precise, *semi-interactive*), composable, information-theoretically secure computation. Our approach is the first to implement several widely used primitives (namely string-commitments, string-OT and sequentially queriable OTMs) at optimal rates. Moreover, our constructions have remarkably low computation complexity, way more efficient than any other construction in the literature. As a drawback, our information-theoretically secure protocols have only limited token reusability, but can be transformed straightforwardly into computationally secure protocols with unlimited token reusability. The computational assumption needed is the weakest standard assumption in cryptography, namely the existence of a pseudorandom number generator, and beyond that we only need the receiver party David to be computationally bounded. After all, our work can be regarded as a substantial gain towards practical two-party computation, but there are still some issues that need some further improvement.

Smaller constants for better practicability. Even though we achieve asymptotically optimal communication complexity, there are some nasty constants left that might make our protocols somewhat slow in practice. In particular, for every l-bit string-OT (or l-bit OTM respectively) the token has to compute and output an $\mathbb{F}_{2^l}^{20\times 5}$-matrix, i.e. we have a blow-up factor of 100. This enormous factor results from two technical artifacts. Firstly, we were only able to prove that our protocol $\Pi_{\text{OAFE}}^{\text{semi-int}}$ securely realizes $\mathbb{F}_q^k$-OAFE, if $k \geq 5$ (cf. Section 8). In contrast, we only need $\mathbb{F}_{2^l}^2$-OAFE for our optimal l-bit string-OT protocol (cf. Section 7.2.1) and there seems to be no potential attack against $\Pi_{\text{OAFE}}^{\text{semi-int}}$ with $k = 2$. Secondly, for technical reasons we need that David chooses a check matrix C of dimension $3k \times 4k$ in step ii of the setup phase (q.v. Figure 7.1) and later computes a check value CW_i from the i-th token output W_i, i.e. we especially need that W_i has dimension $4k \times k$. However, no potential attack is known, if only $C \in \mathbb{F}_q^{\alpha k\times(1+\alpha)k}$ with constant $\alpha > 0$. Now, if we choose $\alpha = \frac{1}{2}$ and $k = 2$, this means that David chooses a check matrix C of dimension 1×3 and the token just needs to compute and output $\mathbb{F}_{2^l}^{3\times 2}$-matrices. In other words, it could be possible to reduce the blow-up factor from 100 to 6 just by more sophisticated proof techniques and a slight modification of the protocol.

Less interaction. Our protocol $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$ (q.v. Figure 7.1) is semi-interactive in the sense that it consists of send and choice phases, such that communication between the sender party Goliath and the receiver party David does only take place in the send phases, whereas Goliath is not involved in the choice phases at all. Moreover, even if Goliath learns all of David's send phase messages in advance (but not before the token is transmitted!), the protocol stays secure. Thus, as David's send phase messages only consist of randomness, we can go with a total of only one single message from David to Goliath, which is sent during the initialization phase of the protocol (cf. Section 7.2.3). However, this approach comes along with two drawbacks. Firstly, the single message from David to Goliath will be quite large. Secondly, David needs to know an upper bound for the number of upcoming send phases, what clearly rules out unlimited token reusability. As a solution for both drawbacks, David could just send a random seed of a pseudorandom number generator. However, it is completely unclear if this does breach security, and how to prove security even if Goliath and the token are computationally bounded.

More realistic hardware assumptions. For security of our protocol $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$ (q.v. Figure 7.1) against a corrupted sender party Goliath we need that the tamper-proof token in David's hands and the token issuer Goliath are perfectly isolated from each other. This assumption is questionable, since one cannot prevent Goliath from placing a very powerful communication device near David's lab. At least, this will enable Goliath to send some messages to the token. However, one can hold the view that the token's transmitting power can be reliably bounded by its weight and size, so that it cannot send any messages back to Goliath. Still, even a unidirectional channel from Goliath to the token suffices to break our protocols.

Therefore, we propose a two-token solution (namely that of Section 7.2.6), where one token just plays Goliath's role of the original protocol. As long as neither token can *send* any message, the tokens are mutually isolated and everything seems well except for one subtle issue: Goliath can change the behavior of the tokens during runtime und thus change his OAFE inputs without being noticed. However, this may be considered unavoidable in real world applications, since a very similar attack could also be mounted if adversarially issued tokens contain clocks.

Closing the gap between primitives and general two-party computation. By our approach we implement OT (and OTMs respectively) via some quite general $\mathbb{F}_q^k$-OAFE functionality (cf. Section 6.4). However, $\mathbb{F}_q^k$-OAFE is strictly stronger than OT in the sense that in general many OT instances and a quite sophisticated protocol are needed to implement $\mathbb{F}_q^k$-OAFE, whereas l-bit string-OT can be implemented rather straightforwardly from a single instance of $\mathbb{F}_2^l$-OAFE or $\mathbb{F}_{2^l}^2$-OAFE (cf. also Section 7.2.1). This raises the question, whether one could base general two-party computation directly on $\mathbb{F}_q^k$-OAFE rather than OT, e.g. via (garbled) arithmetic circuits [Cle91, CFIK03, AIK11], and thereby possibly reduce the computational overhead. More generally, one could also try to implement other sorts of functions directly on the tamper-proof hardware.

Appendices

Table of symbols

General notations

$\mathbb{N}$	naturals including zero; without zero denoted as $\mathbb{N}_{>0}$
$\mathbb{R}$	real numbers; without negative values denoted as $\mathbb{R}_{\geq 0}$; positive values denoted as $\mathbb{R}_{>0}$
$\mathbb{P}[\mathcal{E}]$	probability of the random event $\mathcal{E}$
$\mathbb{E}(\mathbf{x})$	expected value of the random variable $\mathbf{x}$
$\mathbb{H}_2(\mathbf{x})$	collision entropy of the random variable $\mathbf{x}$

Notations in Part I

$\mathrm{pmf}(\Omega)$	set of all probability mass functions over the alphabet Ω
$\mathfrak{F}_{\mathsf{fin}}$	set of all randomized finite 2-party functions; see Secion 2.2 or Notation 3 in Section 4
F	usually an element of $\mathfrak{F}_{\mathsf{fin}}$; cf. Notation 3 in Section 4
$\Upsilon_{\mathrm{A}}, \Upsilon_{\mathrm{B}}$	Alice's/Bob's input alphabet in a randomized finite 2-party function F
$\Omega_{\mathrm{A}}, \Omega_{\mathrm{B}}$	Alice's/Bob's output alphabet in a randomized finite 2-party function F
$\phi_{x,y}(a,b)$	probability of the output tuple (a,b) on input x from Alice and y from Bob
$\eta \in \mathfrak{N}_{\mathrm{B}}^{(F)}$	idealized cheating situation; see Section 3.1.2 and Definition 5 in Section 4.1
$\eta\vert_{\mathrm{A}}(x)$	Alice's input probability of x in the cheating situation η
$\eta\vert_{\mathrm{B}}^{\mathsf{true}}(y)$	Bob's actual input probability of y in the cheating situation η
$\eta\vert_{\mathrm{B}}^{\mathsf{fake}}(y)$	Bob's claimed input probability of y in the cheating situation η
$\xi \in \mathfrak{X}_{\mathrm{B}}^{(F)}$	set of Bob's cheating characteristics for F; see Definition 7 in Section 4.1
$\mathcal{F}_{\mathrm{SMCD}}^{(G,\varepsilon)}$	ideal functionality for correlated data distribution; see Figure 3.4
G	usually a characterization of a correlated data distribution; see Figure 3.4
$\Lambda_{\mathrm{A}}, \Lambda_{\mathrm{B}}$	Alice's/Bob's alphabet of a correlated data distribution G
ψ	probability mass function of a correlated data distribution G
π_F	protocol for generation of correlated data from the 2-party function F; see Figure 4.1
Π_F	set of possible parameter tuples for π_F; see Notation 29 in Section 4.5
ν_{B}	joint distribution of Alice's and Bob's actual and claimed input-output tuples in the protocol step Control A; see Notation 30 in Section 4.5

$\nu_{\mathrm{B}}|_{\mathrm{A}}$ marginal distribution of Alice's inputs in the protocol step Control A; see Notation 30 in Section 4.5

$\nu_{\mathrm{B}}|_{\mathrm{B}}^{\mathsf{true}}$ marginal distribution of Bob's actual inputs in the protocol step Control A; see Notation 30 in Section 4.5

$\nu_{\mathrm{B}}|_{\mathrm{B}}^{\mathsf{fake}}$ marginal distribution of Bob's claimed inputs in the protocol step Control A; see Notation 30 in Section 4.5

$\sim$ equivalence of cheating situations; see Notation 12 in Section 4.3

$\sqsupseteq, \sqsupset$ containedness-relation for cheating situations; see Notation 14 in Section 4.3

$\Psi_F(Y)$ input symbols that Bob can use instead of symbols from Y without being caught cheating; see Notation 21 in Section 4.4

$|s|_\alpha$ number of occurrences of the symbol α in the string s; see Notation 28 in Section 4.5

$s[i]$ i-th element of the string s; see Notation 28 in Section 4.5

$s[K]$ substring of s, indexed by K; see Notation 28 in Section 4.5

$s_{\mathrm{A}} \times s_{\mathrm{B}}$ compound string, whose i-th element is $(s_{\mathrm{A}}[i], s_{\mathrm{B}}[i])$; see Notation 28 in Section 4.5

$\pm$ almost equality; "$a = b \pm c$" stands for "$|a - b| \leq c$"

Notations in Part II

$\mathbb{F}_q$ finite field with q elements

$\mathbb{F}_q^k$ k-dimensional vector space over $\mathbb{F}_q$

$\mathbb{F}_q^{k \times l}$ set of all $(k \times l)$-matrices over $\mathbb{F}_q$

$\mathbf{x} \xleftarrow{\mathrm{r}} X$ the random variable $\mathbf{x}$ is uniformly random over the set X

SD statistical distance; see Section 6.1

ι measure for statistical correlation; see Section 6.1

$\mathcal{F}_{\mathrm{OTM}}$ ideal functionality for one-time memories; see Figure 5.1

$\mathcal{F}_{\mathrm{wrap}}^{\mathrm{stateful}}$ wrapper functionality for modeling tamper-proof hardware; see Figure 6.1

$\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$ ideal functionality for sequential one-time OAFE; see Figure 6.3

$\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$ protocol for implementation of $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$ from a single tamper-proof token; see Figure 7.1

$\mathcal{T}$ tamper-proof token; cf. Figure 7.1

$\mathcal{S}^{\mathsf{David}}$ simulator for a corrupted token receiver David; see Figure 8.1

$\mathcal{S}^{\mathsf{Goliath}}$ simulator for a corrupted token issuer Goliath; see Figure 8.2

Game_l hybrid game in the security proof for $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$; see Section 8.3.6

$\mathcal{F}'$ ideal functionality in Game_l; see Figure 8.3

$\mathcal{S}'_l$ simulator in Game_l; see Figure 8.4

$\mathcal{P}(S)$ power set of S

List of figures

Bibliography

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009.

[AIK11] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. How to garble arithmetic circuits. In Rafail Ostrovsky, editor, *Proceedings of FOCS 2011*, pages 120–129. IEEE, 2011.

[AL07] Yonatan Aumann and Yehuda Lindell. Security against covert adversaries: Efficient protocols for realistic adversaries. In Salil P. Vadhan, editor, *Theory of Cryptography, Proceedings of TCC 2007*, volume 4392 of *Lecture Notes in Computer Science*, pages 137–156. Springer, 2007.

[BBR88] Charles H. Bennett, Gilles Brassard, and Jean-Marc Robert. Privacy amplification by public discussion. *SIAM J. Comput.*, 17(2):210–229, 1988.

[Bea92] Donald Beaver. Foundations of secure interactive computing. In Joan Feigenbaum, editor, *Advances in Cryptology, Proceedings of CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 377–391. Springer, 1992.

[Bea96] Donald Beaver. Correlated pseudorandomness and the complexity of private computations. In *Proceedings of STOC 1996*, pages 479–488. ACM, 1996.

[BKMN09] Julien Brouchier, Tom Kean, Carol Marsh, and David Naccache. Temperature attacks. *IEEE Security & Privacy*, 7(2):79–82, 2009.

[BMM99] Amos Beimel, Tal Malkin, and Silvio Micali. The all-or-nothing nature of two-party secure computation. In Michael J. Wiener, editor, *Advances in Cryptology, Proceedings of CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 80–97. Springer, 1999.

[BOGKW88] Michael Ben-Or, Shafi Goldwasser, Joe Kilian, and Avi Wigderson. Multi-prover interactive proofs: How to remove intractability assumptions. In *Proceedings of STOC 1988*, pages 113–131. ACM, 1988.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings of FOCS 2001*, pages 136–145, 2001. Revised full version online available at `http://eprint.iacr.org/2000/067`.

[CCM98] Christian Cachin, Claude Crépeau, and Julien Marcil. Oblivious transfer with a memory-bounded receiver. In *Proceedings of FOCS 1998*, pages 493–502, 1998.

[CDPW07] Ran Canetti, Yevgeniy Dodis, Rafael Pass, and Shabsi Walfish. Universally composable security with global setup. In Salil P. Vadhan, editor, *Theory of Cryptography, Proceedings of TCC 2007*, volume 4392 of *Lecture Notes in Computer Science*, pages 61–85. Springer, 2007.

[CFIK03] Ronald Cramer, Serge Fehr, Yuval Ishai, and Eyal Kushilevitz. Efficient multi-party computation over rings. In Eli Biham, editor, *Advances in Cryptology, Proceedings of EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 596–613. Springer, 2003.

[CGS08] Nishanth Chandran, Vipul Goyal, and Amit Sahai. New constructions for UC secure computation using tamper-proof hardware. In Nigel P. Smart, editor, *Advances in Cryptology, Proceedings of EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 545–562. Springer, 2008.

[CGT95] Claude Crépeau, Jeroen van de Graaf, and Alain Tapp. Committed oblivious transfer and private multi-party computation. In Don Coppersmith, editor, *Advances in Cryptology, Proceedings of CRYPTO '95*, volume 963 of *Lecture Notes in Computer Science*, pages 110–123. Springer, 1995.

[CK90] Claude Crépeau and Joe Kilian. Weakening security assumptions and oblivious transfer (abstract). In Shafi Goldwasser, editor, *Advances in Cryptology, Proceedings of CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 2–7. Springer, 1990.

[CKS+11] Seung Geol Choi, Jonathan Katz, Dominique Schröder, Arkady Yerukhimovich, and Hong-Sheng Zhou. (Efficient) universally composable two-party computation using a minimal number of stateless tokens. *IACR Cryptology ePrint Archive*, 2011:689, 2011.

[Cle91] Richard Cleve. Towards optimal simulations of formulas by bounded-width programs. *Computational Complexity*, 1:91–105, 1991.

[CMW05] Claude Crépeau, Kirill Morozov, and Stefan Wolf. Efficient unconditional oblivious transfer from almost any noisy channel. In Carlo Blundo and Stelvio Cimato, editors, *SCN 2004*, volume 3352 of *Lecture Notes in Computer Science*, pages 47–59. Springer, 2005.

[Cré88] Claude Crépeau. Equivalence between two flavours of oblivious transfers. In Carl Pomerance, editor, *Advances in Cryptology, Proceedings of CRYPTO '87*, volume 293 of *Lecture Notes in Computer Science*, pages 350–354. Springer, 1988.

[DFR+07] Ivan Damgård, Serge Fehr, Renato Renner, Louis Salvail, and Christian Schaffner. A tight high-order entropic quantum uncertainty relation with applications. In Alfred Menezes, editor, *Advances in Cryptology, Proceedings of CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 360–378. Springer, 2007.

[DKMQ11] Nico Döttling, Daniel Kraschewski, and Jörn Müller-Quade. Unconditional and composable security using a single stateful tamper-proof hardware token. In Yuval Ishai, editor, *Theory of Cryptography, Proceedings of TCC 2011*, volume 6597 of *Lecture Notes in Computer Science*, pages 164–181. Springer, 2011. Extended full version online available at `http://eprint.iacr.org/2012/135`.

[DKMQ12a] Nico Döttling, Daniel Kraschewski, and Jörn Müller-Quade. David & Goliath oblivious affine function evaluation - asymptotically optimal building blocks for universally composable two-party computation from a single untrusted stateful tamper-proof hardware token. *IACR Cryptology ePrint Archive*, 2012:135, 2012. Extended full version of [DKMQ11].

[DKMQ12b] Nico Döttling, Daniel Kraschewski, and Jörn Müller-Quade. Statistically secure linear-rate dimension extension for oblivious affine function evaluation. In Adam Smith, editor, *Information Theoretic Security, Proceedings of ICITS 2012*, volume 7412 of *Lecture Notes in Computer Science*, pages 111–128. Springer, 2012.

[DKS99] Ivan Damgård, Joe Kilian, and Louis Salvail. On the (im)possibility of basing oblivious transfer and bit commitment on weakened security assumptions. In *Advances in Cryptology, Proceedings of EUROCRYPT '99*, pages 56–73, 1999.

[DNW08] Ivan Damgård, Jesper Buus Nielsen, and Daniel Wichs. Isolated proofs of knowledge and isolated zero knowledge. In Nigel P. Smart, editor, *Advances in Cryptology, Proceedings of EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 509–526. Springer, 2008.

[DNW09] Ivan Damgård, Jesper Buus Nielsen, and Daniel Wichs. Universally composable multiparty computation with partially isolated parties. In Omer Reingold, editor, *Theory of Cryptography, Proceedings of TCC 2009*, volume 5444 of *Lecture Notes in Computer Science*, pages 315–331. Springer, 2009.

[EGL85] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 28(6):637–647, 1985.

[FPS+11] Marc Fischlin, Benny Pinkas, Ahmad-Reza Sadeghi, Thomas Schneider, and Ivan Visconti. Secure set intersection with untrusted hardware tokens. In *Proceedings of the 11th international conference on Topics in cryptology: CT-RSA 2011*, CT-RSA'11, pages 1–16, Berlin, Heidelberg, 2011. Springer-Verlag.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 1986.

[GIMS10] Vipul Goyal, Yuval Ishai, Mohammad Mahmoody, and Amit Sahai. Interactive locking, zero-knowledge PCPs, and unconditional cryptography. In Tal Rabin, editor, *Advances in Cryptology, Proceedings of CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 173–190. Springer, 2010.

[GIS+10] Vipul Goyal, Yuval Ishai, Amit Sahai, Ramarathnam Venkatesan, and Akshay Wadia. Founding cryptography on tamper-proof hardware tokens. In Daniele Micciancio, editor, *Theory of Cryptography, Proceedings of TCC 2010*, volume 5978 of *Lecture Notes in Computer Science*, pages 308–326. Springer, 2010.

[GKR08] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. One-time programs. In David Wagner, editor, *Advances in Cryptology, Proceedings of CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 39–56. Springer, 2008.

[GL91] Shafi Goldwasser and Leonid A. Levin. Fair computation of general functions in presence of immoral majority. In Alfred Menezes and Scott A. Vanstone, editors, *Advances in Cryptology, Proceedings of CRYPTO '90*, volume 537 of *Lecture Notes in Computer Science*, pages 77–93. Springer, 1991.

[GO96] Oded Goldreich and Rafail Ostrovsky. Software protection and simulation on oblivious rams. *J. ACM*, 43(3):431–473, 1996.

[Gol04] Oded Goldreich. *Foundations of Cryptography, Volume II: Basic Applications*. Cambridge University Press, 2004.

[HIKN08] Danny Harnik, Yuval Ishai, Eyal Kushilevitz, and Jesper Buus Nielsen. OT-combiners via secure computation. In Ran Canetti, editor, *Theory of Cryptography, Proceedings of TCC 2008*, volume 4948 of *Lecture Notes in Computer Science*, pages 393–411. Springer, 2008.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudo-random generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[HMQU05] Dennis Hofheinz, Jörn Müller-Quade, and Dominique Unruh. Universally composable zero-knowledge arguments and commitments from signature cards. In *Proceedings of the 5th Central European Conference on Cryptology MoraviaCrypt 2005*, 2005.

[HNRR06] Danny Harnik, Moni Naor, Omer Reingold, and Alon Rosen. Completeness in two-party secure computation: A computational view. *Journal of Cryptology*, 19(4):521–552, 2006.

[Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[IKO+11] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, Manoj Prabhakaran, Amit Sahai, and Jürg Wullschleger. Constant-rate oblivious transfer from noisy channels. In Phillip Rogaway, editor, *Advances in Cryptology, Proceedings of CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 667–684. Springer, 2011.

[ILL89] Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-random generation from one-way functions (extended abstract). In *Proceedings of STOC 1989*, pages 12–24. ACM, 1989.

[IPS08] Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Founding cryptography on oblivious transfer - efficiently. In David Wagner, editor, *Advances in Cryptology, Proceedings of CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 572–591. Springer, 2008.

[Kat07] Jonathan Katz. Universally composable multi-party computation using tamper-proof hardware. In Moni Naor, editor, *Advances in Cryptology, Proceedings of EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Computer Science*, pages 115–128. Springer, 2007.

[Kil88] Joe Kilian. Founding cryptography on oblivious transfer. In *Proceedings of STOC 1988*, pages 20–31. ACM, 1988.

[Kil91] Joe Kilian. A general completeness theorem for two-party games. In *Proceedings of STOC 1991*, pages 553–560. ACM, 1991.

[Kil00] Joe Kilian. More general completeness theorems for secure two-party computation. In *Proceedings of STOC 2000*, pages 316–324. ACM, 2000.

[KMQ10] Daniel Kraschewski and Jörn Müller-Quade. Completeness theorems with constructive proofs for finite deterministic 2-party functions (full version). *IACR Cryptology ePrint Archive*, 2010:654, 2010. Full version of [KMQ11].

[KMQ11] Daniel Kraschewski and Jörn Müller-Quade. Completeness theorems with constructive proofs for finite deterministic 2-party functions. In Yuval Ishai, editor, *Theory of Cryptography, Proceedings of TCC 2011*, volume 6597 of *Lecture Notes in Computer Science*, pages 364–381. Springer, 2011. Full version online available at `http://eprint.iacr.org/2010/654`.

[Kol10] Vladimir Kolesnikov. Truly efficient string oblivious transfer using resettable tamper-proof tokens. In Daniele Micciancio, editor, *Theory of Cryptography, Proceedings of TCC 2010*, volume 5978 of *Lecture Notes in Computer Science*, pages 327–342. Springer, 2010.

[Łoj59] Stanisław Łojasiewicz. Sur le problème de la division. *Polska Akademia Nauk. Instytut Matematyczny. Studia Mathematica*, 18:87–136, 1959.

[May95] Dominic Mayers. On the security of the quantum oblivious transfer and key distribution protocols. In Don Coppersmith, editor, *Advances in Cryptology, Proceedings of CRYPTO '95*, volume 963 of *Lecture Notes in Computer Science*, pages 124–135. Springer, 1995.

[May96] Dominic Mayers. Quantum key distribution and string oblivious transfer in noisy channels. In Neal Koblitz, editor, *Advances in Cryptology, Proceedings of CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 343–357. Springer, 1996.

[MPR10] Hemanta K. Maji, Manoj Prabhakaran, and Mike Rosulek. A zero-one law for cryptographic complexity with respect to computational UC security. In Tal Rabin, editor, *Advances in Cryptology, Proceedings of CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 595–612. Springer, 2010.

[MPR12] Hemanta K. Maji, Manoj Prabhakaran, and Mike Rosulek. A unified characterization of completeness and triviality for secure function evaluation. In Steven D. Galbraith and Mridul Nandi, editors, *Progress in Cryptology, Proceedings of INDOCRYPT 2012*, volume 7668 of *Lecture Notes in Computer Science*, pages 40–59. Springer, 2012.

[MPW07] Remo Meier, Bartosz Przydatek, and Jürg Wullschleger. Robuster combiners for oblivious transfer. In Salil P. Vadhan, editor, *Theory of Cryptography, Proceedings of TCC 2007*, volume 4392 of *Lecture Notes in Computer Science*, pages 404–418. Springer, 2007.

[MR92] Silvio Micali and Phillip Rogaway. Secure computation (abstract). In Joan Feigenbaum, editor, *Advances in Cryptology, Proceedings of CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 392–404. Springer, 1992.

[MS08] Tal Moran and Gil Segev. David and Goliath commitments: UC computation for asymmetric parties using tamper-proof hardware. In Nigel P. Smart, editor, *Advances in Cryptology, Proceedings of EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 527–544. Springer, 2008.

[Rab81] Michael O. Rabin. How to exchange secrets by oblivious transfer. Technical report, Aiken Computation Laboratory, Harvard University, 1981.

[Wul07] Jürg Wullschleger. Oblivious-transfer amplification. In Moni Naor, editor, *Advances in Cryptology, Proceedings of EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Computer Science*, pages 555–572. Springer, 2007.

[Wul09] Jürg Wullschleger. Oblivious transfer from weak noisy channels. In Omer Reingold, editor, *Theory of Cryptography, Proceedings of TCC 2009*, volume 5444 of *Lecture Notes in Computer Science*, pages 332–349. Springer, 2009.

[WW06] Stefan Wolf and Jürg Wullschleger. Oblivious transfer is symmetric. In Serge Vaudenay, editor, *Advances in Cryptology, Proceedings of EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 222–232. Springer, 2006.

[Yao82] Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *Proceedings of FOCS 1982*, pages 160–164. IEEE Computer Society, 1982.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *Proceedings of FOCS 1986*, pages 162–167. IEEE Computer Society, 1986.

[Yao95] Andrew Chi-Chih Yao. Security of quantum protocols against coherent measurements. In Frank Thomson Leighton and Allan Borodin, editors, *Proceedings of STOC 1995*, pages 67–75. ACM, 1995.