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

Nico Döttling Daniel Kraschewski Jörn Müller-Quade

Institute of Cryptography and Security, Department of Informatics,
Karlsruhe Institute of Technology, Germany

{doettling,kraschewski,mueller-quade}@kit.edu

Abstract

Cryptographic assumptions regarding tamper-proof hardware tokens have gained increasing attention. Even if the tamper-proof hardware is issued by one of the parties, and hence not necessarily trusted by the other, many tasks become possible: Tamper proof hardware is sufficient for universally composable protocols, for information-theoretically secure protocols, and even allows to create software that can only be used once (one-time programs).

However, all known protocols employing tamper-proof hardware are either indirect, i.e. additional computational assumptions must be used to obtain general two party computations, or a large number of devices must be used. Unfortunately, issuing multiple independent tamper-proof devices requires much stronger isolation assumptions.

This work is the extended version of a recent result of the same authors, where for the first time a protocol was presented that realizes universally composable two party computations (and even one-time programs) with information-theoretic security using only a single tamper-proof device issued by one of the mutually distrusting parties. Now, we present the first protocols for multiple one-time memories (OTMs), and reusable and bidirectional commitment and oblivious transfer (OT) primitives in this setting. All these constructions have only linear communication complexity and are thus asymptotically optimal. Moreover, the computation complexity of our protocols for k-bit OTMs/commitments/OT is dominated by $O(1)$ finite field multiplications with field size 2^k, what is considerably more efficient than any other known construction based on untrusted tamper-proof hardware alone.

The central part of our contribution is a construction for oblivious affine function evaluation (OAFE), which can be seen as a generalization of the well known oblivious transfer primitive: Parametrized by a finite vector space $\mathbb{F}_q^k$, the OAFE primitive allows a designated sender party to choose an arbitrary affine function $f : \mathbb{F}_q \to \mathbb{F}_q^k$, such that hidden from the sender party a designated receiver party may learn $f(x)$ for exactly *one* function argument $x \in \mathbb{F}_q$ of its choice. All our abovementioned results build on this primitive and it may also be of particular interest for the construction of garbled arithmetic circuits.

Keywords: non-interactive secure computation, universal composability, tamper-proof hardware, information-theoretic security, oblivious transfer

Contents

1 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 extend a recent result of ours, where we presented a protocol that realizes universally composable two-party computations (and even one-time programs) with information-theoretic security using only a single (untrusted) tamper-proof device [DKMQ11]. 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.

1.1 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, we suppose that an amplified variant of the protocol of [BOGKW88] can be proven UC-secure.

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 au-

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

thority, to achieve universal composability with respect to global setup assumptions [CDPW07]. 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 k-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 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 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 David.

Query:

- Upon receiving input (x) from 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 1: The ideal/hybrid functionality modeling a single one-time memory (OTM).

Recently, [Kol10] implemented string-OT with stateless tamper-proof hardware tokens, but achieved only covert security. 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 previous 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].

1.2 Our contribution

In this paper 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 supposed 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. For this transformation only a very weak complexity assumption is needed, namely the existence of a pseudorandom number generator (PRNG).

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 2.4). 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, assuming a computationally bounded token receiver our construction allows for unlimited reuse of the tamper-proof hardware, 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 2.4). Last but not least, we achieve a better complexity than in [DKMQ11] (cf. Section 3.2.1 and Section 3.2.3). In particular, by our 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 by that one can implement an arbitrary polynomial number of OTMs 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 (i.e. we 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 lot (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.

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. What is more, our construction allows to 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 respectively. 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 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

	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 2: UC-secure k-bit string-OT based on tamper-proof tokens; table partly taken from [CKS+11]. 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 3.2.7).

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 3.2.7.

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. There are no exponentiations or other operations costlier than finite field multiplication, what makes our constructions remarkably efficient.

1.3 Outline of this paper

The rest of this paper is organized as follows. In Section 2 we introduce some notations (Section 2.1), give a short overview of the notion of security that we use (Section 2.2), describe how our tamper-proof hardware assumption is defined in that framework (Section 2.3) and introduce our new primitive, which we call *sequential one-time OAFE* (Section 2.4). In Section 3.1 we show how one can implement this new primitive from the aforementioned tamper-proof hardware assumption. In Section 3.2 we discuss refinements and some unobvious applications of our construction. At the end of Section 3.2, in Section 3.2.7, 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 4 we give a formal security proof. Finally, in Section 5 we argue for some impossibility results, give a conclusion of our work and suggest directions for improvements and future research.

2 Preliminaries

2.1 Notations

Finite fields, naturals and power sets: By $\mathbb{F}_q$ we denote the finite field of size q. The set of all naturals including zero is denoted by $\mathbb{N}$, without zero it is denoted by $\mathbb{N}_{>0}$. The power set of any set S is denoted by $\mathcal{P}(S)$.

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 $\operatorname{rank}(M) = \operatorname{rank}(C) + k$. 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 $Gx' = y$.

Random events, random variables, support and distribution: We denote random variables by bold face characters, e.g. $\mathbf{x}$, and random events calligraphic, e.g. $\mathcal{E}$. (However, for ease of presentation and better readability, we will sometimes write random variables just like non-random variables, e.g. x). By $\mathbf{x} \in X$ we denote that the support of the random variable $\mathbf{x}$ is some subset of X; in other words, for convenience we sometimes just write $\mathbf{x} \in X$ instead of $\mathbb{P}[\mathbf{x} \in X] = 1$. When $\mathbf{x}$ is uniformly random over X, we denote that by $\mathbf{x} \overset{\mathrm{r}}{\leftarrow} X$.

Probabilities, expected values and collision entropy: We denote the probability that a random variable $\mathbf{x}$ takes some specific value x by $\mathbb{P}[\mathbf{x} = x]$, and analogously for any other relation. If we additionally condition to some event $\mathcal{E}$, we denote the resulting probability by $\mathbb{P}[\mathbf{x} = x \mid \mathcal{E}]$. We denote the expected value of a random variable $\mathbf{x}$ by $\mathbb{E}(\mathbf{x})$ and its collision entropy by $\mathbb{H}_2(\mathbf{x}) = -\log_2 \left(\sum_\alpha \big(\mathbb{P}[\mathbf{x} = \alpha] \big)^2 \right)$.

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

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

When we conditioned the joint distribution of $\mathbf{x}$ and $\mathbf{y}$ to some event $\mathcal{E}$, we denote the resulting statistical distance by $\Delta(\mathbf{x}, \mathbf{y} \mid \mathcal{E})$.

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 $\iota(\mathbf{x}, \mathbf{y}) := \Delta\big((\mathbf{x}, \mathbf{y}), (\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) \big)$ 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.

2.2 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 2.3). Note that for convenience and better readability we use the notation of [Can01] a bit sloppy. E.g., throughout this paper we omit explicit notation of party and session IDs.

2.3 Modeling tamper-proof hardware

2.3.1 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". To model tamper-proof hardware, we employ the $\mathcal{F}_{\text{wrap}}^{\text{stateful}}$ wrapper functionality (q.v. Figure 3). 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 (`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 (`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 3: 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 tokens 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 our results without any computational assumptions, 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.

2.3.2 Real world meaning of our hardware assumption and proof techniques

In Section 4 we will show that our construction from Section 3.1 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 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 is not 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. Furthermore, we argue that all known stateless approaches are subject to attacks that are quite similar to resetting attacks, since they inherently require that the tamper-proof hardware internally generates some (pseudo-)randomness. So as to break security, one just has to make the internal random generator behave predictable (e.g. by freezing the token if physical randomness is used, or by deleting the secret seed of a pseudorandom function). In our view, building stateful tokens secure against resetting attacks is not significantly harder than building stateless tokens secure against side channel attacks that kill the random generator. Anyway, we consider a thorough investigation of this issue an interesting direction for future research.

2.4 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 4). Note that the reductions in Figure 4 also work perfectly for $\mathbb{F}_2^k$-OAFE and k-bit string-OT respectively.

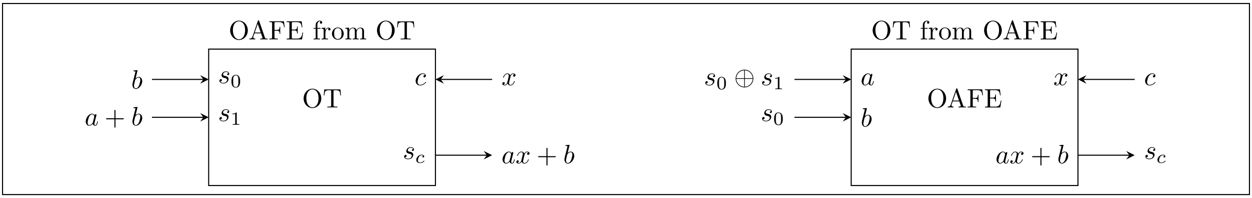


Figure 4: Reductions between bit-OT and $\mathbb{F}_2$-OAFE; protocols taken from [WW06].

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 5.

Note that the reduction protocols in Figure 4 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 5), 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$^+$10] for trusted OTPs from OTMs still works, as there an honest receiver queries all the OTM tokens in a fixed order anyway. Interestingly, the technical challenges dealt with in [GIS$^+$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. Thereto, 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

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

Parametrized by a finite vector space $\mathbb{F}_q^k$, with $k \log q$ being the security parameter, and some runtime bound n that is polynomially bounded in $k \log q$. 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 $(\mathtt{created}, i)$ to the adversary.
- Upon receiving a message $(\mathtt{Delivery}, i)$ 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 $(\mathtt{ready}, i)$ 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 $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$ and the corrupted party, including the ability of deleting and/or forging arbitrary messages.

Figure 5: 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, wheras all messages in the choice phases are delivered immediately.

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.

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

3.1 The basic protocol

We want to implement seq-ot-OAFE (q.v. Section 2.4), 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 a *linear* function (meaning, a polynomial of degree 1) of the corresponding input. However, note that the latter difficulty is only relevant if $q > 2$, as *every* function $\mathbb{F}_2 \to \mathbb{F}_2^k$ is of the form $x \mapsto ax + b$ with constant $(a, b) \in \mathbb{F}_2^k \times \mathbb{F}_2^k$.

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 linear behavior. Otherwise, if the check would be passed for different token inputs $x, x' \in \mathbb{F}_q$ and there do not exist any $r, s \in \mathbb{F}_q^{4k}$ such that $\tau(x) = rx + s$ and $\tau(x') = rx' + s$ 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 $r \xleftarrow{\mathrm{r}} \mathbb{F}_q^{4k}$ and $S \xleftarrow{\mathrm{r}} \mathbb{F}_q^{4k \times k}$.
- Upon receiving the token, David announces a random check matrix $C \xleftarrow{\mathrm{r}} \mathbb{F}_q^{3k \times 4k}$ and a random share $h \xleftarrow{\mathrm{r}} \mathbb{F}_q^{4k} \setminus \{0\}$. 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 $z \xleftarrow{r} \{\tilde{z} \in \mathbb{F}_q^{1\times k} \mid zh = x\}$, 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 $CW \overset{?}{=} \tilde{r}z + \tilde{S}$ 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 $\Pi_{\text{OAFE}}^{\text{semi-int}}$ is given in Figure 6.

There are only two main differences between $\Pi_{\text{OAFE}}^{\text{semi-int}}$ 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 3.2.3 and Section 3.2.5). 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 3.2.2).

At this point we also want to point out that in the protocol description of Figure 6 we purposely do not exactly specify how the parameters k and q depend on the security parameter. In fact, for our security proof we only need that $k \geq 5$; e.g. one can choose k to be constant and q to increase exponentially. With parameters chosen this way, our protocol $\Pi_{\text{OAFE}}^{\text{semi-int}}$ 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 1 below), but for $2 \leq k \leq 4$ we are not aware of any potential attack. However, note that $\mathcal{F}_{\text{OAFE}}^{\text{seq-ot}}$ with $k \leq 4$ 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 $k \leq 4$.

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 3.2.1).

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

Proof. If the sender party Goliath is corrupted, the environment $\mathcal{Z}$ may learn h_1 and h_2—these are the corrupted Goliath's shares of David's first two inputs x_1 and x_2 respectively. As $k = 1$ and hence $h_2 \in \mathbb{F}_q \setminus \{0\}$, the environment $\mathcal{Z}$ can now choose David's second input such that $x_2 = h_1 \cdot h_2$. This way, we get that $z_2 = h_1$ and the token can compute David's first input $x_1 = z_1 \cdot z_2$. Now, the token can abort the protocol or not, depending on the value of x_1. 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, since the simulator does not learn the uncorrupted David's first input x_1. □

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

Parametrized by a finite vector space $\mathbb{F}_q^k$, with $k \log q$ being the security parameter, and some runtime bound n that is polynomially bounded in $k \log q$. 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 $r_i \xleftarrow{\text{r}} \mathbb{F}_q^{4k}$ and a random matrix $S_i \xleftarrow{\text{r}} \mathbb{F}_q^{4k \times k}$, creates a token $\mathcal{T}$ with parameters $(r_1, S_1), \ldots, (r_n, S_n)$ and sends $\mathcal{T}$ to David via $\mathcal{F}_{\text{wrap}}^{\text{stateful}}$. 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 $C \xleftarrow{\text{r}} \mathbb{F}_q^{3k \times 4k}$, 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 $h_i \xleftarrow{\text{r}} \mathbb{F}_q^k \setminus \{0\}$ 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 $z_i \xleftarrow{\text{r}} \{\tilde{z} \in \mathbb{F}_q^{1 \times k} \mid \tilde{z} h_i = x_i\}$ 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 $W_i \notin \mathbb{F}_q^{4k \times k}$, it is treated as an encoding of the all-zero matrix in $\mathbb{F}_q^{4k \times k}$. 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 6: 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.

3.2 Refinements and applications of our construction

Before we give a formal security proof for our protocol $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$, we first want to present how the claimed optimal constructions for multiple OTMs, Commitments and OT (cf. Section 1.2) do work.

As mentioned above, we will prove security of our protocol $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$ only for the case that $k \geq 5$. However, $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$ with $k \leq 4$ 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 for $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$ with $k = 5$ does directly carry over to the case that $k \leq 4$.

At the end of this section, in Section 3.2.7, 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.

3.2.1 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 2.4). However, if one implements some polynomial number (say l) of sequentially queriable OTM tokens by the construction we proposed in Section 2.4, 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 6 in Section 3.1), 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 $\big((\tilde{r}_1, \tilde{S}_1, 1), (\tilde{a}_1, \tilde{b}_1, 1), \ldots, (\tilde{r}_l, \tilde{S}_l, l), (\tilde{a}_l, \tilde{b}_l, l)\big)$. 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 $\big((\tilde{r}_1, \tilde{S}_1), \ldots, (\tilde{r}_n, \tilde{S}_n)\big)$ 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 3.2.3) or commitments (cf. Section 3.2.5) 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. Theorem 1 in [DKMQ11]. Furthermore, our new two-round protocol is an improvement upon [DKMQ11], where we needed four rounds of communication between David and Goliath.

3.2.2 Computational solution for unlimited reusability of a memory-limited token

Our protocol $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$ (q.v. Figure 6 in Section 3.1) guarantees perfect security against David (cf. Section 4.2). 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 $\Pi_{\text{OAFE}}^{\text{semi-int}}$ and the ideal functionality $\mathcal{F}_{\text{OAFE}}^{\text{seq-ot}}$ 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.

3.2.3 Unidirectional string-OT and OTMs with optimal communication complexity

As discussed in Section 2.4, 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 6 in Section 2.4), 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; 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.

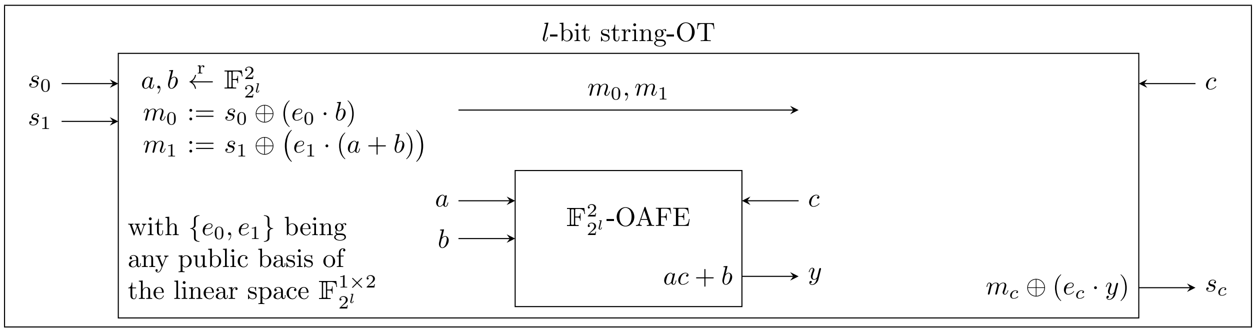

l-bit string-OT
s_0
s_1
$a, b \xleftarrow{r} \mathbb{F}_{2^l}^2$
$m_0 := s_0 \oplus (e_0 \cdot b)$
$m_1 := s_1 \oplus \big(e_1 \cdot (a + b)\big)$
m_0, m_1
c
with $\{e_0, e_1\}$ being any public basis of the linear space $\mathbb{F}_{2^l}^{1\times 2}$
a
b
$\mathbb{F}_{2^l}^2$-OAFE
c
$ac + b$
y
$m_c \oplus (e_c \cdot y)$
s_c

Figure 7: 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 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 2.4). 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 beats all known protocols for OT (or OTMs respectively) based on untrusted tamper-proof hardware in the literature.

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 modular exponentiation.

3.2.4 Achieving optimal communication complexity for bidirectional string-OT

In Section 3.2.3 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 $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$ 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 8 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 only later on input some $a \in \mathbb{F}_q$ of his choice into the underlying $\mathbb{F}_q$-OAFE instance (and then compute $b = m - z$). However, in our case this problem can be solved straightforwardly: Since our protocol $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$ 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 2.4).

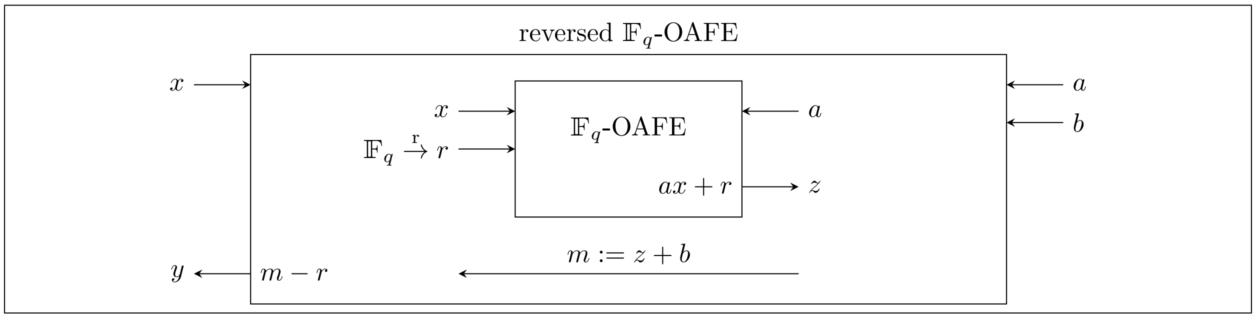
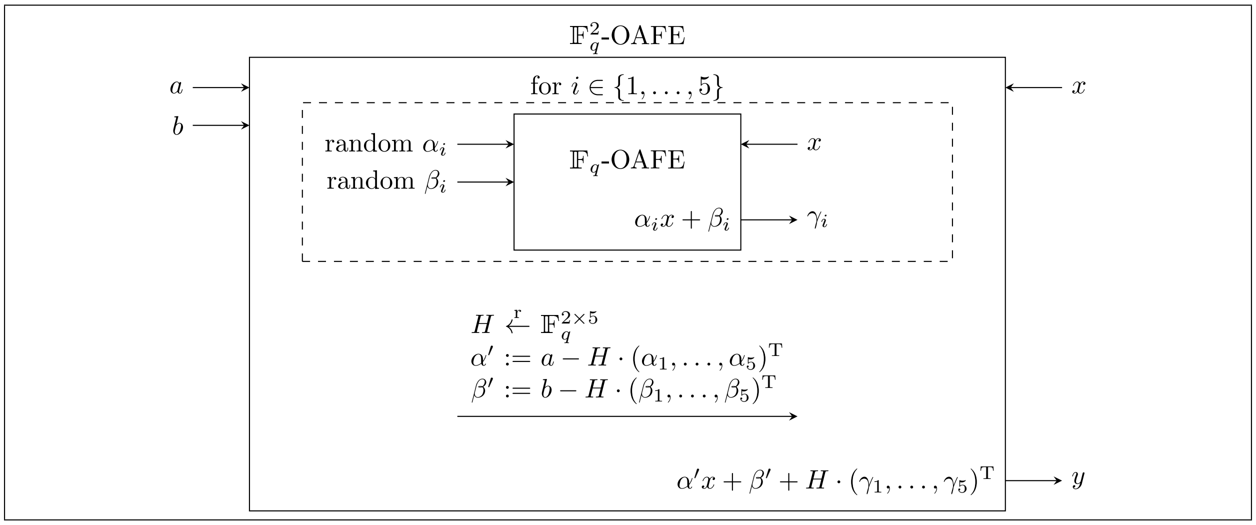


Figure 8: Basic approach for reversing the direction of a given $\mathbb{F}_q$-OAFE primitive; protocol taken from [WW06]. Note that this protocol is not UC-secure, unless input of a into the underlying $\mathbb{F}_q$-OAFE instance is enforced before the receiver outputs y; otherwise a corrupted sender can maliciously delay his choice of a (and b).

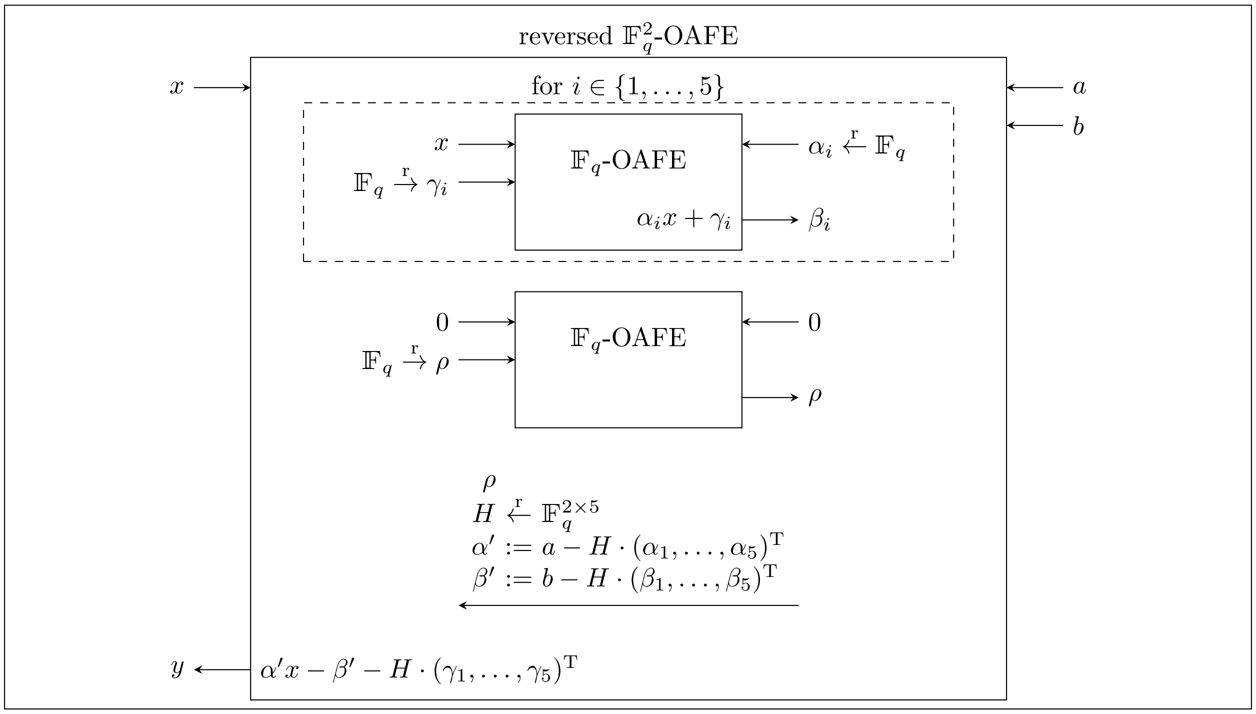
Obviously, the approach in Figure 8 does not work for $\mathbb{F}_q^k$-OAFE with $k > 1$, but we need $\mathbb{F}_q^2$-OAFE for our aimed at OT protocol. Thus, a construction for $\mathbb{F}_q^k$-OAFE from a $O(k)$ instances of $\mathbb{F}_q$-OAFE would come in very handy. In [DKMQ12] one can find such 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 9 the approach of [DKMQ12] with $k = 2$. By combining this with the protocol in Figure 8 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 10.

Now, by plugging the protocol of Figure 10 on top of $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$ (q.v. Figure 6 in Section 3.1) 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 in Section 3.2.3 and thereby implement string-OT with optimal communication complexity also for the direction from the token receiver to the token issuer.



$\mathbb{F}_q^2$-OAFE
a
b
x
for $i \in \{1,\dots,5\}$
random α_i
random β_i
$\mathbb{F}_q$-OAFE
x
$\alpha_i x + \beta_i$
γ_i
$H \xleftarrow{r} \mathbb{F}_q^{2\times 5}$
$\alpha' := a - H \cdot (\alpha_1,\dots,\alpha_5)^{\mathrm{T}}$
$\beta' := b - H \cdot (\beta_1,\dots,\beta_5)^{\mathrm{T}}$
$\alpha' x + \beta' + H \cdot (\gamma_1,\dots,\gamma_5)^{\mathrm{T}}$
y

Figure 9: Implementation of $\mathbb{F}_q^2$-OAFE from five instances of $\mathbb{F}_q$-OAFE; protocol taken from [DKMQ12]. For UC-security it is required that (H, α', β') is not announced before the receiver party has provided some input to all five underlying $\mathbb{F}_q$-OAFE instances in the dashed box.



reversed $\mathbb{F}_q^2$-OAFE
x
a
b
for $i \in \{1,\dots,5\}$
x
$\mathbb{F}_q \xrightarrow{r} \gamma_i$
$\mathbb{F}_q$-OAFE
$\alpha_i \xleftarrow{r} \mathbb{F}_q$
$\alpha_i x + \gamma_i$
β_i
0
$\mathbb{F}_q \xrightarrow{r} \rho$
$\mathbb{F}_q$-OAFE
0
ρ
ρ
$H \xleftarrow{r} \mathbb{F}_q^{2\times 5}$
$\alpha' := a - H \cdot (\alpha_1,\dots,\alpha_5)^{\mathrm{T}}$
$\beta' := b - H \cdot (\beta_1,\dots,\beta_5)^{\mathrm{T}}$
y
$\alpha' x - \beta' - H \cdot (\gamma_1,\dots,\gamma_5)^{\mathrm{T}}$

Figure 10: 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.

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

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 $b_i \overset{\mathrm{r}}{\leftarrow} \mathbb{F}_{2^l}$ 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 $x_i \overset{\mathrm{r}}{\leftarrow} \mathbb{F}_{2^l}$. 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 $\Pi_{\mathrm{COM}}^{\mathrm{backward}}$

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 $a_i, b_i, c_i, d_i \overset{\mathrm{r}}{\leftarrow} \mathbb{F}_{2^l}$ 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 11: 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.

3.2.5 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 11 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.

3.2.6 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 6 in Section 2.4). 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. Theorem 1 in [DKMQ11]. However, we can easily give a 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 11) can be made non-interactive, as Goliath does not need to derandomize any of his inputs. All other protocols become non-interactive straightforwardly.

3.2.7 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 12). 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 2 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 2. The protocols $\Pi_{\text{COM}}^{\text{enlarge}}$ and $\Pi_{\text{OTM}}^{\text{enlarge}}$ in Figure 12 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 $1 - 2^{l-k}$. $\square$

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 $\tilde{s}_0, \tilde{s}_1 \overset{\mathrm{r}}{\leftarrow} \{0,1\}^l$ 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 $(\mathtt{Commit}, c)$ 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 $\tilde{s} \overset{\mathrm{r}}{\leftarrow} \{0,1\}^l$, 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 $(\mathtt{committed})$.

Unveil phase:

3. Upon input $(\mathtt{Unveil})$ 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 $\tilde{s}_0, \tilde{s}_1 \overset{\mathrm{r}}{\leftarrow} \{0,1\}^l$, 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 $(\mathtt{ready})$.

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 12: 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.

4 Correctness and security of our protocol

In this section we show that in the $\mathcal{F}_{\mathrm{wrap}}^{\mathrm{stateful}}$-hybrid model our protocol $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$ is a universally composable implementation of the ideal functionality $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{semi-int}}$, if only $k \geq 5$. In particular, in Section 4.2 we will prove perfect security against a corrupted David for all k and in Section 4.3 we will prove statistical security against a corrupted Goliath for the case that $k \geq 5$. However, first of all we will show that $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$ always works correctly when no party is corrupted (Section 4.1).

4.1 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:

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

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

4.2 Security against a corrupted receiver

We first show security against a corrupted receiver, as this is the easy case. The respective simulator is depicted in Figure 13 and the high level idea of how simulation works is as follows.

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 then can query the ideal functionality $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$ on this input x_i, thus receiving y_i, and then just revise $\tilde{b}_i$ so that $y_i = G W_i h_i + \tilde{a}_i x_i + \tilde{b}_i$.

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 revises the token's output W_i so that the check $C W_i \overset{?}{=} \tilde{r}_i z_i + \tilde{S}_i$ is still passed, but GW 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 then can query the ideal functionality $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$ on this input x_i, thus receiving y_i, and then revise W_i by some W' so that $CW' = CW_i$ and $y_i = G W' h_i + \tilde{a}_i x_i + \tilde{b}_i$. Note that existence of such W' is always guaranteed, since G is complementary to C (i.e. especially G has full rank) and $h \neq 0$. Now, by Theorem 3, we give our formal security proof.

Theorem 3. *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 $\mathcal{F}_{\mathrm{OAFE}}^{\mathrm{seq-ot}}$ and simulator $\mathcal{S}^{\mathrm{David}}(\mathcal{A})$ is identically distributed to the view of $\mathcal{Z}$ in the real model with protocol $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$ and adversary $\mathcal{A}$.*

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

- Set up an honest Goliath-machine $\mathcal{G}$; 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 David). Wire the simulated machines $\mathcal{A}, \mathcal{G}, \mathcal{F}_{\text{wrap}}^{\text{stateful}}$ to each other and $\mathcal{A}$ to the environment right the way they would be wired in the real model.
- Upon receiving a message $(\texttt{created}, i)$ from the ideal functionality $\mathcal{F}_{\text{OAFE}}^{\text{seq-ot}}$, reply with $(\texttt{Delivery}, i)$. Then, upon receiving $(\texttt{ready}, i)$ on behalf of the corrupted David, choose some random vectors $a_i, b_i \xleftarrow{\text{r}} \mathbb{F}_q^k$ 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 a send phase, extract the current state j of j'_{queried} from the view of the simulated $\mathcal{F}_{\text{wrap}}^{\text{stateful}}$. 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 $\mathcal{F}_{\text{wrap}}^{\text{stateful}}$.
  1. Compute $x_i \leftarrow z_i h_i$ and (on behalf of the corrupted David) send (x_i, i) to the ideal functionality $\mathcal{F}_{\text{OAFE}}^{\text{seq-ot}}$; let (y_i, i) denote the respective answer from $\mathcal{F}_{\text{OAFE}}^{\text{seq-ot}}$.
  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, extract the current state i of j'_{queried} from the view of the simulated $\mathcal{F}_{\text{wrap}}^{\text{stateful}}$. When $\mathcal{G}$ has already received (and not ignored) a message (h_i, i) in step 3 of a send phase, 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 $\mathcal{F}_{\text{wrap}}^{\text{stateful}}$.
  1. Compute $x_i \leftarrow z_i h_i$ and (on behalf of the corrupted David) send (x_i, i) to the ideal functionality $\mathcal{F}_{\text{OAFE}}^{\text{seq-ot}}$; let (y_i, i) denote the respective answer from $\mathcal{F}_{\text{OAFE}}^{\text{seq-ot}}$.
  2. Choose randomly $W' \xleftarrow{\text{r}} \{\tilde{W} \in \mathbb{F}_q^{4k \times k} \mid C\tilde{W} = \tilde{r}_i z_i + \tilde{S}_i \ \wedge \ G\tilde{W}h_i + \tilde{a}_i x_i + \tilde{b}_i = y_i\}$.

Figure 13: The simulator program $\mathcal{S}^{\text{David}}(\mathcal{A})$, given an adversary $\mathcal{A}$ that corrupts the receiver David.

Proof. Let the random variable $\mathbf{view}_{\text{real}}^{\mathcal{A}}(\mathcal{Z})$ denote the view of $\mathcal{Z}$ in the real model with adversary $\mathcal{A}$ and let $\mathbf{view}_{\text{ideal}}^{\mathcal{A}}(\mathcal{Z})$ denote the view of $\mathcal{Z}$ in the ideal model with simulator $\mathcal{S}^{\text{David}}(\mathcal{A})$, i.e. we have to show that the statistical distance $\Delta\big(\mathbf{view}_{\text{real}}^{\mathcal{A}}(\mathcal{Z}), \mathbf{view}_{\text{ideal}}^{\mathcal{A}}(\mathcal{Z})\big)$ is zero. W.l.o.g. we assume that $\mathcal{Z}$ has a hard coded random tape that maximizes this statistical distance. Thereby, w.l.o.g. we have that in step ii of the setup phase the corrupted David announces some fixed $(\hat{C}, \hat{G}) \in \mathbb{F}_q^{3k \times 4k} \times \mathbb{F}_q^{k \times 4k}$ with $\hat{G}$ complementary to $\hat{C}$. Now we can represent the ideal model by the following ensemble of random variables $(i = 1, \dots, n)$:

- $(\mathbf{r}_i, \mathbf{S}_i) \xleftarrow{\text{r}} \mathbb{F}_q^{4k} \times \mathbb{F}_q^{4k \times k}$ represents the i-th initialization parameter of the token.
- $\hat{\mathbf{a}}_i, \hat{\mathbf{b}}_i \in \mathbb{F}_q^k$ represent the ideal Goliath's i-th input from $\mathcal{Z}$.
- $\mathbf{a}_i, \mathbf{b}_i \xleftarrow{\text{r}} \mathbb{F}_q^k$ represent the random input of the simulated machine $\mathcal{G}$ in the i-th send phase.
- $\tilde{\mathbf{r}}_i := \hat{C}\mathbf{r}_i$ and $\tilde{\mathbf{S}}_i := \hat{C}\mathbf{S}_i$ represent the announcement of $\mathcal{G}$ in step 1 of the i-th send phase.
- $\mathbf{h}_i \in \mathbb{F}_q^k \setminus \{0\}$ represents the corrupted David's announcement in step 2 of the i-th send phase.

- $\tilde{\mathbf{a}}_i, \tilde{\mathbf{b}}_i \in \mathbb{F}_q^k$ represent the announcement of $\mathcal{G}$ in step 3 of the i-th send phase.
- $\mathbf{z}_i \in \mathbb{F}_q^{1\times k}$ represents the corrupted David's i-th input into the token.
- $\mathbf{x}_i := \mathbf{z}_i\mathbf{h}_i$ represents the simulator's i-th input to the ideal functionality $\mathcal{F}_{\text{wrap}}^{\text{stateful}}$.
- $\mathbf{y}_i := \hat{\mathbf{a}}_i\mathbf{x}_i + \hat{\mathbf{b}}_i$ represents the ideal functionality's i-th output to the simulator $\mathcal{S}^{\text{David}}(\mathcal{A})$.
- $\mathbf{W}_i \in \mathbb{F}_q^{4k\times k}$ represents the token's i-th output.

When the corrupted David has announced $\mathbf{h}_i$ before he queried the token with input $(\mathbf{z}_i, i)$, the simulator will not overwrite the simulated Goliath's announcement $(\tilde{\mathbf{a}}_i, \tilde{\mathbf{b}}_i, i)$, but the token's output $\mathbf{W}_i$ will be overwritten. We will refer to this as the *regular* case. In this case it holds:

$$\begin{aligned}
\tilde{\mathbf{a}}_i &\leftarrow \mathbf{a}_i - \hat{G}\mathbf{r}_i \\
\tilde{\mathbf{b}}_i &\leftarrow \mathbf{b}_i - \hat{G}\mathbf{S}_i\mathbf{h}_i \\
\mathbf{W}_i &\xleftarrow{\text{r}} \left\{\tilde{W} \in \mathbb{F}_q^{4k\times k} \;\middle|\; \hat{C}\tilde{W} = \tilde{\mathbf{r}}_i\mathbf{z}_i + \tilde{\mathbf{S}}_i \;\wedge\; \hat{G}\tilde{W}\mathbf{h}_i + \tilde{\mathbf{a}}_i\mathbf{x}_i + \tilde{\mathbf{b}}_i = \mathbf{y}_i\right\}
\end{aligned}$$

When the corrupted David has queried the token with input $(\mathbf{z}_i, i)$ before he announced $\mathbf{h}_i$, the simulator will not overwrite the token's output $\mathbf{W}_i$, but the simulated Goliath's announcement $(\tilde{\mathbf{a}}_i, \tilde{\mathbf{b}}_i, i)$ will be overwritten. We will refer to this as the *irregular* case. In this case it holds:

$$\begin{aligned}
\mathbf{W}_i &\leftarrow \mathbf{r}_i\mathbf{z}_i + \mathbf{S}_i \\
\tilde{\mathbf{a}}_i &\leftarrow \mathbf{a}_i - \hat{G}\mathbf{r}_i \\
\tilde{\mathbf{b}}_i &\leftarrow \mathbf{y}_i - \hat{G}\mathbf{W}_i\mathbf{h}_i - \tilde{\mathbf{a}}_i\mathbf{x}_i \qquad \text{i.e. } \tilde{\mathbf{b}}_i = \hat{\mathbf{b}}_i - \hat{G}\mathbf{S}_i\mathbf{h}_i + (\hat{\mathbf{a}}_i - \mathbf{a}_i)\mathbf{x}_i
\end{aligned}$$

Note that we did not exactly specify the distributions of the $\mathbf{h}_i, \mathbf{z}_i, \hat{\mathbf{a}}_i, \hat{\mathbf{b}}_i$, as these values are chosen by the corrupted David or the environment $\mathcal{Z}$ respectively and may depend on all information that $\mathcal{Z}$ has gathered so far. As we fixed the random tape of $\mathcal{Z}$, the random variable $\mathbf{view}_{\text{ideal}}^{\mathcal{A}}(\mathcal{Z})$ becomes a function value of $(\tilde{\mathbf{r}}_1, \tilde{\mathbf{S}}_1, \tilde{\mathbf{a}}_1, \tilde{\mathbf{b}}_1, \mathbf{W}_1, \ldots, \tilde{\mathbf{r}}_n, \tilde{\mathbf{S}}_n, \tilde{\mathbf{a}}_n, \tilde{\mathbf{b}}_n, \mathbf{W}_n)$. Now, let $\mathcal{E}$ denote the event that $\forall i \in \{1,\ldots,n\} : (\mathbf{a}_i, \mathbf{b}_i) = (\hat{\mathbf{a}}_i, \hat{\mathbf{b}}_i)$ and let $\mathcal{E}'$ denote the event that $\forall i \in \{1,\ldots,n\} : \mathbf{W}_i = \mathbf{r}_i\mathbf{z}_i + \mathbf{S}_i$. The rest of our indistinguishability proof will proceed as follows: We first show that conditioning the random variable $\mathbf{view}_{\text{ideal}}^{\mathcal{A}}(\mathcal{Z})$ to the event $\mathcal{E}$ does not change its distribution, then we show that further conditioning $\mathbf{view}_{\text{ideal}}^{\mathcal{A}}(\mathcal{Z})$ to the event $(\mathcal{E} \wedge \mathcal{E}')$ still does not change its distribution. However, this will already conclude our proof, as $\mathbf{view}_{\text{ideal}}^{\mathcal{A}}(\mathcal{Z})$ conditioned to $(\mathcal{E} \wedge \mathcal{E}')$ clearly is identically distributed to $\mathbf{view}_{\text{real}}^{\mathcal{A}}(\mathcal{Z})$. Note that the events $\mathcal{E}$ and $\mathcal{E}'$ both have non-zero probability, as for each $i \in \{1,\ldots,n\}$ the random variables $\mathbf{a}_i$ and $\mathbf{b}_i$ are independent of $(\hat{\mathbf{a}}_i, \hat{\mathbf{b}}_i)$ and in the case that $(\mathbf{a}_i, \mathbf{b}_i) = (\hat{\mathbf{a}}_i, \hat{\mathbf{b}}_i)$ it always holds that $\hat{C}(\mathbf{r}_i\mathbf{z}_i + \mathbf{S}_i) = \tilde{\mathbf{r}}_i\mathbf{z}_i + \tilde{\mathbf{S}}_i$ and $\hat{G}(\mathbf{r}_i\mathbf{z}_i + \mathbf{S}_i)\mathbf{h}_i + \tilde{\mathbf{a}}_i\mathbf{x}_i + \tilde{\mathbf{b}}_i = \mathbf{y}_i$.

In our first step we have to show that conditioning to $\mathcal{E}$ does not alter the joint distribution of $\tilde{\mathbf{r}}_1, \tilde{\mathbf{S}}_1, \tilde{\mathbf{a}}_1, \tilde{\mathbf{b}}_1, \mathbf{W}_1, \ldots, \tilde{\mathbf{r}}_n, \tilde{\mathbf{S}}_n, \tilde{\mathbf{a}}_n, \tilde{\mathbf{b}}_n, \mathbf{W}_n$. Let us consider some arbitrary but fixed $i \in \{1,\ldots,n\}$. Now note that in the regular case $\tilde{\mathbf{a}}_i$ and $\tilde{\mathbf{b}}_i$ are the only random variables that are computed from $(\mathbf{a}_i, \mathbf{b}_i)$. Furthermore, in the regular case $\tilde{\mathbf{a}}_i$ and $\tilde{\mathbf{b}}_i$ also are the only random variables that are computed from $\hat{G}\mathbf{r}_i$ and $\hat{G}\mathbf{S}_i$; all other values computed from $\mathbf{r}_i$ and $\mathbf{S}_i$ in fact are computed from $\hat{C}\mathbf{r}_i$ and $\hat{C}\mathbf{S}_i$. Thus, since $\hat{G}$ is complementary to $\hat{C}$ and $\mathbf{r}_i, \mathbf{S}_i$ are uniformly random (and $\mathbf{h}_i \neq 0$), we just could have stated $\tilde{\mathbf{a}}_i, \tilde{\mathbf{b}}_i \xleftarrow{\mathrm{r}} \mathbb{F}_q^k$ instead of $\tilde{\mathbf{a}}_i \leftarrow \mathbf{a}_i - \hat{G}\mathbf{r}_i$ and $\tilde{\mathbf{b}}_i \leftarrow \mathbf{b}_i - \hat{G}\mathbf{S}_i\mathbf{h}_i$, whereby independence of $(\tilde{\mathbf{r}}_i, \tilde{\mathbf{S}}_i, \tilde{\mathbf{a}}_i, \tilde{\mathbf{b}}_i, \mathbf{W}_i)$ from $(\mathbf{a}_i, \mathbf{b}_i)$ becomes obvious. In the irregular case things are even easier, as there $\tilde{\mathbf{a}}_i$ is computed the same way as in the regular case, but $\tilde{\mathbf{b}}_i$ does not depend on $\mathbf{b}_i$. Since i was arbitrary, it follows that the joint distribution of $\tilde{\mathbf{r}}_1, \tilde{\mathbf{S}}_1, \tilde{\mathbf{a}}_1, \tilde{\mathbf{b}}_1, \mathbf{W}_1, \dots, \tilde{\mathbf{r}}_n, \tilde{\mathbf{S}}_n, \tilde{\mathbf{a}}_n, \tilde{\mathbf{b}}_n, \mathbf{W}_n$ is independent of $(\mathbf{a}_1, \mathbf{b}_1, \dots, \mathbf{a}_n, \mathbf{b}_n)$ and we can henceforth condition all our considerations to the event $\mathcal{E}$ without changing the joint distribution of $\tilde{\mathbf{r}}_1, \tilde{\mathbf{S}}_1, \tilde{\mathbf{a}}_1, \tilde{\mathbf{b}}_1, \mathbf{W}_1, \dots, \tilde{\mathbf{r}}_n, \tilde{\mathbf{S}}_n, \tilde{\mathbf{a}}_n, \tilde{\mathbf{b}}_n, \mathbf{W}_n$.

In our second step we have to show that conditioning to $\mathcal{E}'$ does not alter the joint distribution of $\tilde{\mathbf{r}}_1, \tilde{\mathbf{S}}_1, \tilde{\mathbf{a}}_1, \tilde{\mathbf{b}}_1, \mathbf{W}_1, \dots, \tilde{\mathbf{r}}_n, \tilde{\mathbf{S}}_n, \tilde{\mathbf{a}}_n, \tilde{\mathbf{b}}_n, \mathbf{W}_n$, when already conditioned to $\mathcal{E}$. Again, let us consider some arbitrary but fixed $i \in \{1, \dots, n\}$. Since in the irregular case we have that $\mathbf{W}_i = \mathbf{r}_i\mathbf{z}_i + \mathbf{S}_i$ by construction, we can focus to the regular case. There, since conditioned to the event that $(\mathbf{a}_i, \mathbf{b}_i) = (\hat{\mathbf{a}}_i, \hat{\mathbf{b}}_i)$, we have that $\mathbf{W}_i \xleftarrow{\mathrm{r}} \big\{\tilde{W} \in \mathbb{F}_q^{4k\times k} \mid \hat{C}\tilde{W} = \hat{C}(\mathbf{r}_i\mathbf{z}_i + \mathbf{S}_i) \ \wedge\ \hat{G}\tilde{W}\mathbf{h}_i = \hat{G}(\mathbf{r}_i\mathbf{z}_i + \mathbf{S}_i)\mathbf{h}_i\big\}$. This is equivalent to randomly sampling $\mathbf{S}' \xleftarrow{\mathrm{r}} \big\{\tilde{S}' \in \mathbb{F}_q^{4k\times k} \mid \hat{C}\tilde{S}' = \hat{C}\mathbf{S}_i \ \wedge\ \hat{G}\tilde{S}'\mathbf{h}_i = \hat{G}\mathbf{S}_i\mathbf{h}_i\big\}$ and then just setting $\mathbf{W}_i \leftarrow \mathbf{r}_i\mathbf{z}_i + \mathbf{S}'$. However, in the regular case the joint distribution of $\tilde{\mathbf{r}}_i, \tilde{\mathbf{S}}_i, \tilde{\mathbf{a}}_i, \tilde{\mathbf{b}}_i, \mathbf{W}_i$ is identical to the joint distribution of $\tilde{\mathbf{r}}_i, \hat{C}\mathbf{S}', \tilde{\mathbf{a}}_i, \tilde{\mathbf{b}}_i, \mathbf{W}_i$ by construction, what concludes our proof. □

4.3 Security against a corrupted sender

The case of a corrupted sender Goliath is the technically challenging part of the security proof. The respective simulator is depicted in Figure 14.

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. 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 $u_i \xleftarrow{\mathrm{r}} \mathbb{F}_q^{1\times k}$ and input (u_i, i) into the token $\mathcal{T}$; let W_i denote the token's output (w.l.o.g. $W_i \in \mathbb{F}_q^{4k\times k}$). 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; else just set $f_i \leftarrow \top$.
  2. Pick a random vector $v_i \xleftarrow{\mathrm{r}} \mathbb{F}_q^{1\times k}$ and input (v_i, i) into a copy of $\mathcal{T}'$; let W' denote the token's output (w.l.o.g. $W' \in \mathbb{F}_q^{4k\times k}$). 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 $(\bot, i)$ 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 $r_i \in \mathbb{F}_q^{4k}$, such that $W_i - W' = r_i(u_i - v_i)$, and set $S_i \leftarrow W_i - r_i u_i$. Then compute $a_i \leftarrow \tilde{a}_i + G r_i$ and $b_i \leftarrow \tilde{b}_i + G S_i h_i$.
  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 14: The simulator program $\mathcal{S}^{\text{Goliath}}(\mathcal{A})$, given an adversary $\mathcal{A}$ that corrupts the sender party.

The high level picture of how this simulator works is as follows. The send phases are simulated perfectly, as $\mathcal{Z}$ just interacts with a simulated version of the complete real model. However, 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. Thereto, the simulator first checks whether the token acts honestly in the i-th choice phase. He does so by just executing the i-th choice phase with 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 a linear function that describes the token behavior in this phase. Once having extracted this linear 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 $k \log q$, 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 5.2 for a further discussion on this issue.

Lemma 4. *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 $\mathcal{T}$.*

Proof. We just have to show that the expected number of iterations in step 2 of the simulator's extraction program (q.v. Figure 14) is polynomially bounded. In fact, we will even show that the expected number of iterations is constant. 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, we had the following probability that exactly t iterations are performed:

$$\begin{array}{rl} 1-p & \text{for } t = 0 \\ p^2 \cdot (1-p)^{t-1} & \text{for } t > 0 \end{array}$$

This yields the following upper bound for the expected number of iterations:

$$p^2 \cdot \sum\nolimits_{t=1}^{\infty} t \cdot (1-p)^{t-1}$$

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:

$$p \cdot \sum\nolimits_{t=1}^{\infty} t \cdot (1-p)^{t-1} \;=\; \tfrac{1}{p}$$

Putting things together, we have shown that the expected number of iterations is at most 1. □

4.3.1 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 statistically indistinguishable

Functionality $\mathcal{F}'$

Parametrized by a finite vector space $\mathbb{F}_q^k$, with $k \log q$ being the security parameter, and some runtime bound n that is polynomially bounded in $k \log q$. 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 $\mathcal{F}_{\text{OAFE}}^{\text{seq-ot}}$ and the corrupted party, including the ability of deleting and/or forging arbitrary messages.

Figure 15: The ideal functionality for the hybrid games $\mathsf{Game}_0, \ldots, \mathsf{Game}_n$.

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 15. Each simulator $\mathcal{S}'_l(\mathcal{A})$ overwrites the first l outputs, so that with overwhelming probability they equal the first l David outputs in the real model. The remaining $n - l$ outputs are computed from an extracted linear 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 16. By the next lemma and the subsequent corollary we show indistinguishability between Game_0 and the ideal model, and between Game_n and the real model respectively. The corresponding proofs make use of some technical lemmata, which we moved to a separate section (Section 4.3.3).

Lemma 5. *Let some arbitrary environment $\mathcal{Z}$ be given and some adversary $\mathcal{A}$ that corrupts the sender Goliath. Then the probability that the simulator $\mathcal{S}'_l(\mathcal{A})$ in Game_l gives up (cf. step 3 in the simulator's extraction program in Figure 16) is upper bounded by:*

$$2n \left((1+q)q^{1-k} + \sqrt{\exp\bigl(lq^{2-k}\bigr) - 1} \right)$$

Proof. For arbitrary but fixed $i \in \{1, \ldots, n\}$ let us consider the situation that $\mathcal{S}'_l(\mathcal{A})$ enters his extraction program for the i-th time and that $f_{i-1} = \top$. We have to estimate the probability that $f_i = \top$, but $v_i = w_i$ or any row of the matrix $\tau_i(w_i) - \tau_i(v_i)$ is linearly independent of $v_i - w_i$. Thereto, let V_i denote the set of token inputs that pass David's consistency check and let Z_i denote the support of w_i, i.e. we have:

$$V_i = \left\{ \tilde{v} \in \mathbb{F}_q^{1\times k} \;\middle|\; C \cdot \tau_i(\tilde{v}) = \tilde{r}_i \tilde{v} + \tilde{S}_i \right\} \qquad Z_i = \begin{cases} \{\tilde{z} \in \mathbb{F}_q^{1\times k} \mid \tilde{z} h_i = x_i\} & \text{if } i \leq l \\ \mathbb{F}_q^{1\times k} & \text{else} \end{cases}$$

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 $\tau_i(\tilde{v})$; w.l.o.g. $\tau_i(\tilde{v}) \in \mathbb{F}_q^{4k\times k}$ for all $\tilde{v} \in \mathbb{F}_q^{1\times k}$. 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 run the following extraction program:
  1. Pick two random vectors $u_i \xleftarrow{r} \mathbb{F}_q^{1\times k}$ and $z_i \xleftarrow{r} \{\tilde{z} \in \mathbb{F}_q^{1\times k} \mid \tilde{z}h_i = x_i\}$. If $i \le l$, set $w_i \leftarrow z_i$; else set $w_i \leftarrow u_i$. Then input (w_i, i) into the token $\mathcal{T}$, thus progressing its internal state.
  2. If $f_{i-1} = \bot$ or $C \cdot \tau_i(w_i) \neq \tilde{r}_i w_i + \tilde{S}_i$, set $f_i \leftarrow \bot$ and go to step 4. Otherwise set $f_i \leftarrow \top$ and pick a random vector $v_i \xleftarrow{r} \{\tilde{v} \in \mathbb{F}_q^{1\times k} \mid C \cdot \tau_i(\tilde{v}) = \tilde{r}_i \tilde{v} + \tilde{S}_i\}$.
  3. If $v_i = w_i$ or $\tau_i(w_i) - \tau_i(v_i) \neq r(w_i - v_i)$ for all $r \in \mathbb{F}_q^k$, give up and (i.e., send $(\bot, i)$ to the environment and terminate). Otherwise compute the unique vector $r_i \in \mathbb{F}_q^{1\times k}$, such that $\tau_i(w_i) - \tau_i(v_i) = r(w_i - v_i)$, and set $S_i \leftarrow \tau_i(w_i) - r_i w_i$. Then compute $a_i \leftarrow \tilde{a}_i + G r_i$ and $b_i \leftarrow \tilde{b}_i + G S_i h_i$.
  4. If $f_i = \top$, compute $y_i \leftarrow a_i x_i + b_i$; else just set $y_i \leftarrow 0$. Then send $(\texttt{Reply}, y_i, i)$ to $\mathcal{F}'$.

Figure 16: 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 statistically indistinguishable from the real model with adversary $\mathcal{A}$ and the ideal model with simulator $\mathcal{S}^{\text{Goliath}}(\mathcal{A})$ respectively.

Thereby, when we arbitrarily fix $(\tau_i, C, \tilde{r}_i, \tilde{S}_i, h_i)$ and thus (V_i, Z_i), we can already estimate the probability that $f_i = \top$ and $v_i = w_i$ as follows:

$$\mathbb{P}[f_i = \top \;\wedge\; v_i = w_i] \;=\; \mathbb{P}[w_i \in V_i \;\wedge\; v_i = w_i] \;=\; \tfrac{|Z_i \cap V_i|}{|Z_i|} \cdot \tfrac{1}{|V_i|} \;\le\; \tfrac{1}{|Z_i|} \;\le\; q^{1-k}$$

Now we estimate the probability that $f_i = \top$ and at the same time $\tau_i(w_i) - \tau_i(v_i) \neq r(v_i - w_i)$ for all $r \in \mathbb{F}_q^k$. By the definition of ι (see Section 2.1) and the technical Lemma 13 we can estimate:

$$\mathbb{P}\Big[\exists\, W, W' \in \tau_i\big(\mathbb{F}_q^{1\times k}\big) :\ \operatorname{rank}\big(W - W'\big) \ge 1 > \operatorname{rank}\big(C(W - W')\big)\Big] \;<\; q^{1-3k}|\mathbb{F}_q^{1\times k}|^2 + \iota(C, \tau_i)$$
$$\mathbb{P}\Big[\exists\, W, W' \in \tau_i\big(\mathbb{F}_q^{1\times k}\big) :\ \operatorname{rank}\big(W - W'\big) \ge 2 > \operatorname{rank}\big(C(W - W')\big)\Big] \;<\; q^{2-3k}|\mathbb{F}_q^{1\times k}|^2 + \iota(C, \tau_i)$$

In other words, at least with probability $1 - q^{1-k} - q^{2-k} - 2\cdot\iota(C, \tau_i)$ we have the following implications for all $\tilde{v}, \tilde{v}' \in \mathbb{F}_q^{1\times k}$:

$$\operatorname{rank}\big(\tau_i(\tilde{v}) - \tau_i(\tilde{v}')\big) > 0 \;\Rightarrow\; \operatorname{rank}\big(C \cdot \tau_i(\tilde{v}) - C \cdot \tau_i(\tilde{v}')\big) > 0$$
$$\operatorname{rank}\big(\tau_i(\tilde{v}) - \tau_i(\tilde{v}')\big) > 1 \;\Rightarrow\; \operatorname{rank}\big(C \cdot \tau_i(\tilde{v}) - C \cdot \tau_i(\tilde{v}')\big) > 1$$

Now we can apply Lemma 14, whereby we get:

$$\mathbb{P}\big[|V_i| \le q \;\vee\; \exists\, r \in \mathbb{F}_q^{4k},\, S \in \mathbb{F}_q^{4k\times k}\, \forall\, \tilde{v} \in V_i :\ \tau_i(\tilde{v}) = r\tilde{v} + S\big] \;\ge\; 1 - (1+q)q^{1-k} - 2\cdot\iota(C, \tau_i)$$

However, on the one hand we have that $\mathbb{P}\big[f_i = \top \;\big|\; |V_i| \le q\big] \le q^{2-k}$, as one verifies quite easily. On the other hand, if there exist $r \in \mathbb{F}_q^{4k}$ and $S \in \mathbb{F}_q^{4k\times k}$, such that $\tau_i(\tilde{v}) = r\tilde{v} + S$ for all $\tilde{v} \in V_i$, then it can not be that $w_i \in V_i$ and at the same time $\tau_i(w_i) - \tau_i(v_i) \ne r(w_i - v_i)$ for all $r \in \mathbb{F}_q^{4k}$. Putting things together, we have shown:

$$\mathbb{P}\big[f_i = \top \;\wedge\; \forall r \in \mathbb{F}_q^{4k} : \tau_i(w_i) - \tau_i(v_i) \ne r(w_i - v_i)\big] \;\le\; (1+2q)q^{1-k} + 2\cdot\iota(C,\tau_i)$$

So, finally we have to estimate $\iota(C,\tau_i)$. Note that w.l.o.g. the environment in Game_l plays deterministically and hence David's input x_i can be written as a function value $x_i(C, h_1, \ldots, h_n, w_1, \ldots w_{i-1})$. Further note that $\iota(C,\tau_i) \le \iota\big(C,(w_1,\ldots,w_{i-1})\big)$, since τ_i solely depends on $(w_1,\ldots,w_{i-1})$. Also note that $\iota\big(C,(w_1,\ldots,w_{i-1})\big) = \iota\big(C,(w_1,\ldots,w_l)\big)$ if $i > l$, since $(w_{l+1},\ldots,w_n) = (u_{l+1},\ldots,u_n)$ and thus $(w_{l+1},\ldots,w_n)$ is just completely independent randomness. By Corollary 20 we can conclude:

$$\iota(C,\tau_i) \;<\; \begin{cases} \sqrt{\exp\big((i-1)q^{2-k}\big) - 1} & \text{if } i \le l \\ \sqrt{\exp\big(lq^{2-k}\big) - 1} & \text{if } i > l \end{cases}$$

All in all, the probability that $\mathcal{S}'_l(\mathcal{A})$ does give up during the i-th execution of his extraction program can be estimated as follows:

$$\mathbb{P}\Big[f_i = \top \wedge \Big(v_i = w_i \vee \forall r \in \mathbb{F}_q^{4k} : \tau_i(w_i) - \tau_i(v_i) \ne r(w_i - v_i)\Big)\Big] \;<\; 2(1+q)q^{1-k} + 2\sqrt{\exp\big(lq^{2-k}\big) - 1}$$

Our lemma now just follows by the Union Bound. □

Corollary 6. *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 $\mathcal{S}^{\mathrm{Goliath}}(\mathcal{A})$ and the view of $\mathcal{Z}$ in the hybrid game Game_0 are statistically close in $k \log q$. Furthermore, the view of $\mathcal{Z}$ in the real model with protocol $\Pi_{\mathrm{OAFE}}^{\mathrm{semi-int}}$ and adversary $\mathcal{A}$ and the view of $\mathcal{Z}$ in the hybrid game Game_n also are statistically close in $k \log q$, if only $k \ge 3$.*

Proof. We first show that the ideal model and Game_0 are indistinguishable. There are only two differences between the ideal model and Game_0. Firstly, they differ in the exact moment when the simulator runs his extraction program: The simulator $\mathcal{S}^{\mathrm{Goliath}}(\mathcal{A})$ runs his extraction program directly after the simulated David machine $\mathcal{D}$ has finished a send phase, whereas $\mathcal{S}'_0(\mathcal{A})$ runs his extraction program as recently as the ideal David queries the functionality $\mathcal{F}'$ in a choice phase. However, this is completely invisible to the environment. The only other difference between the ideal model and Game_0 is that $\mathcal{S}^{\mathrm{Goliath}}(\mathcal{A})$ does give up after q^k iterations in step 2 of his extraction program. Thus, it suffices to show that $\mathcal{S}^{\mathrm{Goliath}}(\mathcal{A})$ does reach the limit of q^k iterations only with negligible probability. However, this is a direct consequence of Lemma 4.

Now we show that the real model and Game_n are indistinguishable, if only $k \ge 3$. Note that, if $f_i = \top$, we can write David's output y_i in Game_n as follows:

$$y_i = a_i x_i + b_i = (\tilde{a}_i + G r_i)x_i + \tilde{b}_i + G\underbrace{\big(\tau_i(z_i) - r_i z_i\big)}_{=S_i} h_i = G_i \cdot \tau_i(z_i) \cdot h_i + \tilde{a}_i x_i + \tilde{b}_i + G r_i \underbrace{(x_i - z_i h_i)}_{=0}$$

Thus, the environment's view in Game_n may only differ from its view in the real model, if $\mathcal{S}'_n(\mathcal{A})$ does give up in step 3 of his extraction program. However, by Lemma 5 we know that this may happen only with negligible probability, if only $k \ge 3$. □

4.3.2 Indistinguishability of successive hybrid games

In this section we finally show indistinguishability between the hybrid games $\mathsf{Game}_0, \dots, \mathsf{Game}_n$. Our proof is game-based, i.e. we define four games $\Gamma_1(\mathbb{F}_q^k, n, l), \dots, \Gamma_4(\mathbb{F}_q^k, n, l)$ and show:

- The maximal winning probability (or to be more precise, the maximal advantage over a purely random strategy) in $\Gamma_1(\mathbb{F}_q^k, n, l)$ induces a meaningful upper bound for the statistical distance between the environment's view in Game_{l-1} and its view in Game_l (Lemma 7).
- The maximal winning probability in $\Gamma_{i+1}(\mathbb{F}_q^k, n, l)$ is only negligibly higer than the maximal winning probability in $\Gamma_i(\mathbb{F}_q^k, n, l)$ (Lemmata 8, 9 and 10).
- The maximal winning probability in $\Gamma_4(\mathbb{F}_q^k, n, l)$ is only negligibly higher than the winning probability of a purely random strategy (Proposition 11), what is sufficient to conclude our indistinguishability proof (Corollary 12).

The corresponding proofs make use of some technical lemmata, which we moved to a separate section (Section 4.3.3).

The high level idea of our proof strategy is as follows. The game $\Gamma_1(\mathbb{F}_q^k, n, l)$ is just a straightforward abstraction of Game_{l-1} and Game_l. Any distinguishing environment for Game_{l-1} and Game_l can be straightforwardly transformed into a winning player for $\Gamma_1(\mathbb{F}_q^k, n, l)$. The change from $\Gamma_1(\mathbb{F}_q^k, n, l)$ to $\Gamma_2(\mathbb{F}_q^k, n, l)$ corresponds to changing the simulator $\mathcal{S}'_l(\mathcal{A})$ (q.v. Figure 16) such that the extracted token parameters (r_i, S_i) no longer depend on the check matrix C. As now the extracted token parameters (r_i, S_i) solely depend on token inputs, we can next change the token functionality such that it directly outputs the parameters (r_i, S_i). This is done by $\Gamma_3(\mathbb{F}_q^k, n, l)$. In $\Gamma_3(\mathbb{F}_q^k, n, l)$ the player/adversary still has to correctly forecast a linear projection (Cr_i, CS_i) of the token parameters (r_i, S_i) or the game is aborted. However, if the linear mapping defined by C has no collisions on any possible token parameters (r_i, S_i) and (r'_i, S'_i) of the current stage, the player/adversary could as well forecast the extracted token parameters directly. This is what he has to do in $\Gamma_4(\mathbb{F}_q^k, n, l)$. Now, the player/adversary has to correctly forecast everything he might learn through token outputs and we can compute an upper bound for the amount of this information.

Lemma 7. *Let $\frac{1}{2} + \delta$ be the maximal winning probability in the game $\Gamma_1(\mathbb{F}_q^k, n, l)$. Then the statistical distance between the environment's view in Game_{l-1} and its view in Game_l is upper bounded by 2δ.*

Proof. Let some environment $\mathcal{Z}$ and some adversary $\mathcal{A}$ be given and let ε denote the statistical distance between the views of $\mathcal{Z}$ in the hybrid games Game_{l-1} and Game_l with simulator $\mathcal{S}'_{l-1}(\mathcal{A})$ and $\mathcal{S}'_l(\mathcal{A})$ respectively. It is straightforward to see, how the player $\mathcal{K}$ in the game $\Gamma_1(\mathbb{F}_q^k, n, l)$ can generate a view of $\mathcal{Z}$ in Game_{l-d}, where d is the secret random bit that $\mathcal{K}$ tries to guess. The high level idea can be sketched as follows:

- The mappings $\tau_1, \dots, \tau_n$ are specified by the token $\mathcal{T}$; when we fix the first $i-1$ token inputs $w_1, \dots, w_{i-1}$, the token functionality in the i-th choice phase is $w_i \mapsto \tau_i(w_1, \dots, w_i)$.
- The variables $C, h_1, \dots, h_l, (\tilde{r}_1, \tilde{S}_1, x_1, w_1, v_1, r_1, S_1), \dots, (\tilde{r}_n, \tilde{S}_n, x_n, w_n, v_n, r_n, S_n)$ correspond to their same-named counterparts in Game_{l-d}. Note that the player $\mathcal{K}$ can choose $h_{l+1}, \dots, h_n$ arbitrarily and w.l.o.g. $\mathcal{Z}$ and $\mathcal{A}$ ignore these values, since they are just uniform randomness and statistically independent of everything else in Game_{l-1} as well as in Game_l.

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 $\mathcal{H} := \mathbb{F}_q^k \setminus \{0\}$ and $\mathcal{C} := \mathbb{F}_q^{3k\times 4k}$ and $\mathcal{M} := \mathbb{F}_q^{4k} \times \mathbb{F}_q^{4k\times k}$. The player $\mathcal{K}$ is computationally unbounded.

1. For each $i \in \{1, \dots, n\}$ 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 l random vectors $h_1, \dots, h_l \overset{\mathrm{r}}{\leftarrow} \mathcal{H}$ and a random matrix $C \overset{\mathrm{r}}{\leftarrow} \mathcal{C}$.
3. A random bit $d \overset{\mathrm{r}}{\leftarrow} \{0, 1\}$ is chosen secretly.
4. For $i = 1, \dots, n$:
   (a) The player $\mathcal{K}$ chooses some $\tilde{r}_i \in \mathbb{F}_q^{3k}$ and $\tilde{S}_i \in \mathbb{F}_q^{3k\times k}$ and $x_i \in \mathbb{F}_q$.
   (b) If $i \leq l - d$, let $w_i := z_i \overset{\mathrm{r}}{\leftarrow} \{\tilde{z} \in \mathcal{U} \mid \tilde{z}h_i = x_i\}$, else let $w_i := u_i \overset{\mathrm{r}}{\leftarrow} \mathcal{U}$, chosen secretly.
   (c) If $C \cdot \tau_i(w_1, \dots, w_i) \neq \tilde{r}_i w_i + \tilde{S}_i$, the player $\mathcal{K}$ is notified about that by a special message $(*, i)$ and the game is aborted in the sense that step 5 follows next. Otherwise, a random vector $v_i \overset{\mathrm{r}}{\leftarrow} \big\{\tilde{v} \in \mathcal{U} \;\big|\; C \cdot \tau_i(w_1, \dots, w_{i-1}, \tilde{v}) = \tilde{r}_i\tilde{v} + \tilde{S}_i\big\}$ is chosen secretly.
   (d) If $v_i = w_i$ or $\tau_i(w_1, \dots, w_i) - \tau_i(w_1, \dots, w_{i-1}, v_i) \neq r(w_i - v_i)$ for all $r \in \mathbb{F}_q^k$, the player $\mathcal{K}$ is notified about that by a special message $(\perp, i)$ and the game is aborted in the sense that step 5 follows next; else $\mathcal{K}$ learns the unique tuple $(r_i, S_i) \in \mathcal{M}$, such that $\tau_i(w_1, \dots, w_i) = r_i w_i + S_i$ and $\tau_i(w_1, \dots, w_{i-1}, v_i) = r_i v_i + S_i$.
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 17: Definition of a stand-alone indistinguishability game that captures the difference between the hybrid games Game_{l-1} and Game_l.

- When the game $\Gamma_1(\mathbb{F}_q^k, n, l)$ is aborted in the i-th round of step 4 and the abort happens in substep 4c, this corresponds to the event that $f_i = \perp$ and hence $y_i = \ldots = y_n = 0$.
- When the game $\Gamma_1(\mathbb{F}_q^k, n, l)$ is aborted in the i-th round of step 4 and the abort happens in substep 4d, this corresponds to the event that the simulator $\mathcal{S}'_{l-d}(\mathcal{A})$ does give up and outputs $(\perp, i)$.

It is straightforward to see that this way the player $\mathcal{K}$ can perfectly emulate a view of $\mathcal{Z}$ in the hybrid game Game_{l-d}. Now, let the random variable $\mathbf{view}_\mathcal{Z}$ denote this emulated view and let the random variable $\mathbf{d}$ denote the secret random bit that $\mathcal{K}$ tries to guess. Since $\mathcal{K}$ may win the game $\Gamma_1(\mathbb{F}_q^k, n, l)$ 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 exists some 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:

$$dist_l \;=\; \underbrace{\mathbb{P}\big[P(\mathbf{view}_\mathcal{Z}) = 1 \;\big|\; \mathbf{d} = 1\big]}_{=2\mathbb{P}[P(\mathbf{view}_\mathcal{Z})=1 \wedge \mathbf{d}=1]} - \Big(1 - \underbrace{\mathbb{P}\big[P(\mathbf{view}_\mathcal{Z}) = 0 \;\big|\; \mathbf{d} = 0\big]}_{=2\mathbb{P}[P(\mathbf{view}_\mathcal{Z})=0 \wedge \mathbf{d}=0]}\Big) \;\leq\; 2\delta \qquad \square$$

Game $\Gamma_2(\mathbb{F}_q^k, n, l)$

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 $\mathcal{H} := \mathbb{F}_q^k \setminus \{0\}$ and $\mathcal{C} := \mathbb{F}_q^{3k\times 4k}$ and $\mathcal{M} := \mathbb{F}_q^{4k} \times \mathbb{F}_q^{4k\times k}$. The player $\mathcal{K}$ is computationally unbounded.

1. For each $i \in \{1, \dots, n\}$ 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 l random vectors $h_1, \dots, h_l \overset{\text{r}}{\leftarrow} \mathcal{H}$ and a random matrix $C \overset{\text{r}}{\leftarrow} \mathcal{C}$.
3. A random bit $d \overset{\text{r}}{\leftarrow} \{0, 1\}$ is chosen secretly.
4. For $i = 1, \dots, n$:
   (a) The player $\mathcal{K}$ chooses some $\tilde{r}_i \in \mathbb{F}_q^{3k}$ and $\tilde{S}_i \in \mathbb{F}_q^{3k\times k}$ and $x_i \in \mathbb{F}_q$.
   (b) If $i \leq l - d$, let $w_i := z_i \overset{\text{r}}{\leftarrow} \{\tilde{z} \in \mathcal{U} \mid \tilde{z}h_i = x_i\}$, else let $w_i := u_i \overset{\text{r}}{\leftarrow} \mathcal{U}$, chosen secretly.
   (c) If there exists a unique tuple $(r_i, S_i) \in \mathcal{M}$, such that $\tau_i(w_1, \dots, w_i) = r_i w_i + S_i$ and $\#\big\{\tilde{v} \in \mathcal{U} \mid \tau_i(w_1, \dots, w_{i-1}, \tilde{v}) = r_i\tilde{v} + S_i\big\} \geq q^{3k/4}$, and for this unique tuple it holds that $(Cr_i, CS_i) = (\tilde{r}_i, \tilde{S}_i)$, then the player $\mathcal{K}$ learns (r_i, S_i); else the player $\mathcal{K}$ receives a special message $(*, i)$ 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 18: The second version of our indistinguishability game. The only difference from $\Gamma_1(\mathbb{F}_q^k, n, l)$ is that the messages for the player $\mathcal{K}$ in step 4 are now computed deterministically.

Lemma 8. *Let some finite vector space $\mathbb{F}_q^k$ and $n, l \in \mathbb{N}_{>0}$ be given, such that $l \leq n$ and $q^k \geq 22$, i.e. $q^k \leq q^{9k/8} - q^{3k/4}$. Let $\frac{1}{2} + \delta$ be the maximal winning probability in the game $\Gamma_2(\mathbb{F}_q^k, n, l)$. Then the maximal winning probability in the game $\Gamma_1(\mathbb{F}_q^k, n, l)$ is upper bounded by:*

$$\tfrac{1}{2} + \delta + n\left(2q^{1-k/4} + 3(1+2q)q^{1-k}\right) + 4\sum_{i=0}^{n-1}\sqrt{\exp(iq^{2-k}) - 1}$$

Proof. The only difference between $\Gamma_1(\mathbb{F}_q^k, n, l)$ and $\Gamma_2(\mathbb{F}_q^k, n, l)$ is that in step 4c of the game $\Gamma_2(\mathbb{F}_q^k, n, l)$ the message for the player $\mathcal{K}$ is computed deterministically, rather than probabilistically as in $\Gamma_1(\mathbb{F}_q^k, n, l)$. So all we have to show is that in the game $\Gamma_1(\mathbb{F}_q^k, n, l)$ with sufficiently high probability the message for $\mathcal{K}$ in the steps 4c and 4d is the same as if it were computed the deterministic way of $\Gamma_2(\mathbb{F}_q^k, n, l)$.

Thereto, for arbitrary but fixed $i \in \{1, \dots, n\}$ we consider the i-th round of step 4 in $\Gamma_1(\mathbb{F}_q^k, n, l)$. Given that the game is not aborted before this round, we show:

1. If the game is aborted and $\mathcal{K}$ receives the message $(*, i)$, then there does not exist any tuple $(r, S) \in \mathcal{M}$, such that $\tau_i(w_1, \dots, w_i) = rw_i + S$ and $(Cr, CS) = (\tilde{r}_i, \tilde{S}_i)$. In other words, the player $\mathcal{K}$ would have received the same message in the game $\Gamma_2(\mathbb{F}_q^k, n, l)$.
2. Only with very little probability it may happen that the player $\mathcal{K}$ receives the message $(\perp, i)$. The probability is upper bounded by $2\left((1+q)q^{1-k} + \sqrt{\exp\big((i-1)q^{2-k}\big) - 1}\right)$.
3. If the player $\mathcal{K}$ receives some $(r_i, S_i) \in \mathcal{M}$, then we have that $\tau_i(w_1, \dots, w_i) = r_i w_i + S_i$.

4. If the player $\mathcal{K}$ receives some $(r_i, S_i) \in \mathcal{M}$, then we have that $(Cr_i, CS_i) = (\tilde{r}_i, \tilde{S}_i)$.

5. Only with very little probability it may happen that there exist two or more tuples $(r, S) \in \mathcal{M}$, such that $\tau_i(w_1, \ldots, w_i) = rw_i + S$ and $\#\{\tilde{v} \in \mathcal{U} \mid \tau_i(w_1, \ldots, w_{i-1}, \tilde{v}) = r\tilde{v} + S\} \geq q^{3k/4}$. The probability is upper bounded by $q^{1-k/4}$.

6. Only with very little probability it may happen that the player $\mathcal{K}$ receives some $(r_i, S_i) \in \mathcal{M}$, such that $\#\{\tilde{v} \in \mathcal{U} \mid \tau_i(w_1, \ldots, w_{i-1}, \tilde{v}) = r_i\tilde{v} + S_i\} < q^{3k/4}$. The probability is upper bounded by $q^{1-k/4} + (1 + 2q)q^{1-k} + 2\sqrt{\exp\big((i-1)q^{2-k}\big) - 1}$.

Putting things together, we will have that with very high probability the player $\mathcal{K}$ does receive the same message as if it were computed by step 4c of the game $\Gamma_2(\mathbb{F}_q^k, n, l)$. In particular, by adding the above bounds we get $2q^{1-k/4} + 3(1 + 2q)q^{1-k} + 4\sqrt{\exp\big((i-1)q^{2-k}\big) - 1}$ as an upper bound for the probability that $\mathcal{K}$ receives a different message. Given this upper bound, our lemma follows by the Union Bound.

Assertion 1 can be shown straightforwardly. If there exits some tuple $(r, S) \in \mathcal{M}$, such that $\tau_i(w_1, \ldots, w_i) = rw_i + S$ and $(Cr, CS) = (\tilde{r}_i, \tilde{S}_i)$, then we have that $C \cdot \tau(w_1, \ldots, w_i) = C(rw_i + S) = \tilde{r}_i w_i + \tilde{S}_i$ and hence $\mathcal{K}$ does not receive the message $(*, i)$.

Assertion 2 needs some more sophisticated arguments; see below.

Assertion 3 holds trivially.

Assertion 4 can be shown very straightforwardly, exploiting the following facts. If the player $\mathcal{K}$ receives some $(r_i, S_i) \in \mathcal{M}$, we have that $v_i \neq w_i$ and we have the following equation system:

$$\begin{aligned} C \cdot \tau_i(w_1, \ldots, w_i) &= \tilde{r}_i w_i + \tilde{S}_i & \qquad \tau_i(w_1, \ldots, w_i) &= r_i w_i + S_i \\ C \cdot \tau_i(w_1, \ldots, w_{i-1}, v_i) &= \tilde{r}_i v_i + \tilde{S}_i & \qquad \tau_i(w_1, \ldots, w_{i-1}, v_i) &= r_i v_i + S_i \end{aligned}$$

Thus, $Cr_i(w_i - v_i) = \tilde{r}_i(w_i - v_i)$ and $C(r_i w_i + S_i) = \tilde{r}_i w_i + \tilde{S}_i$. However, since $v_i \neq w_i$, this implies that $Cr_i = \tilde{r}_i$ and $CS_i = \tilde{S}_i$.

Assertion 5 is a direct consequence of Lemma 18. Note that $q^k \leq q^{9k/8} - q^{3k/4}$ by assumption. So once $w_1, \ldots, w_{i-1}$ are fixed, we have by Lemma 18 that there are less than $q^{3k/4}$ different values for w_i such that $\tau_i(w_1, \ldots, w_i) = rw_i + S$ and $\#\{\tilde{v} \in \mathcal{U} \mid \tau_i(w_1, \ldots, w_{i-1}, \tilde{v}) = r\tilde{v} + S\} \geq q^{3k/4}$. Thus, the probability that w_i takes one of these values is upper bounded by $\frac{q^{3k/4}}{q^k - 1}$.

Assertion 6 can be shown by the same arguments as assertion 2; see below.

For our proof of the assertions 2 and 6 we need a more formal treatment of what happens in $\Gamma_1(\mathbb{F}_q^k, n, l)$. So, let any player $\mathcal{K}$ be given. W.l.o.g. we assume that $\mathcal{K}$ plays deterministically and hence the mappings $\tau_1, \ldots, \tau_n$ are fixed. We define the following random variables that just represent the respective random values in $\Gamma_1(\mathbb{F}_q^k, n, l)$ with player $\mathcal{K}$: $\mathbf{h}_1, \ldots, \mathbf{h}_l \in \mathcal{H}$, $\mathbf{C} \in \mathcal{C}$, $\mathbf{d} \in \{0, 1\}$, $\mathbf{w}_1, \ldots, \mathbf{w}_n, \mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathcal{U} \cup \{\perp\}$, $(\tilde{\mathbf{r}}_1, \tilde{\mathbf{S}}_1, \ldots, \tilde{\mathbf{r}}_n, \tilde{\mathbf{S}}_n) \in (\mathbb{F}_q^{4k} \times \mathbb{F}_q^{4k \times k}) \cup \{(\perp, \perp)\}$ and $(\mathbf{r}_1, \mathbf{S}_1), \ldots, (\mathbf{r}_n, \mathbf{S}_n) \in \mathcal{M} \cup \{(\perp, \perp)\}$. If the game is aborted, we represent that by the event that all remaining random variables are set to $\perp$. Moreover, we define the random variables

$\mathbf{W}_i := \tau_i(\mathbf{w}_1, \ldots, \mathbf{w}_{i-1}, \mathcal{U})$ and $\mathbf{V}_i := \{\tilde{v} \in \mathcal{U} \mid \mathbf{C} \cdot \tau_i(\mathbf{w}_1, \ldots, \mathbf{w}_{i-1}, \tilde{v}) = \tilde{\mathbf{r}}_i \tilde{v} + \tilde{\mathbf{S}}_i\}$. Let $\mathcal{E}$ denote the event that $(\mathbf{r}_i, \mathbf{S}_i) \neq (\perp, \perp)$, i.e. the game is not aborted in stage i or earlier. We have to bound the following probabilities:

$$\text{assertion 2:} \quad \mathbb{P}\Big[\mathbf{w}_i \in \mathbf{V}_i \ \wedge \ \big(\mathbf{v}_i = \mathbf{w}_i \ \vee \ \overbrace{\forall (r, S) \in \mathcal{M} \, \exists \, \tilde{v} \in \mathbf{V}_i : \ \tau_i(\mathbf{w}_1, \ldots, \mathbf{w}_{i-1}, \tilde{v}) \neq r\tilde{v} + S}^{\mathcal{E}' :=}\big)\Big]$$

$$\text{assertion 6:} \quad \mathbb{P}\Big[\mathcal{E} \ \wedge \ \overbrace{\#\{\tilde{v} \in \mathcal{U} \mid \tau_i(\mathbf{w}_1, \ldots, \mathbf{w}_{i-1}, \tilde{v}) = \mathbf{r}_i \tilde{v} + \mathbf{S}_i\} < q^{3k/4}}^{\mathcal{E}'' :=}\Big]$$

Note that by the definition of ι (see Section 2.1) and Lemma 13 we have:

$$\mathbb{P}\Big[\exists \, W, W' \in \mathbf{W}_i : \ \operatorname{rank}\big(W - W'\big) \geq 1 > \operatorname{rank}\big(C(W - W')\big)\Big] \ < \ q^{1-3k}|\mathcal{U}|^2 + \iota(\mathbf{C}, \mathbf{W}_i)$$
$$\mathbb{P}\Big[\exists \, W, W' \in \mathbf{W}_i : \ \operatorname{rank}\big(W - W'\big) \geq 2 > \operatorname{rank}\big(C(W - W')\big)\Big] \ < \ q^{2-3k}|\mathcal{U}|^2 + \iota(\mathbf{C}, \mathbf{W}_i)$$

In other words, with probability higher than $1 - (1+q)q^{1-k} - 2 \cdot \iota(\mathbf{C}, \mathbf{W}_i)$ the following implications hold true for all $W, W' \in \mathbf{W}_i$:

$$\operatorname{rank}(W - W') > 0 \ \Rightarrow \ \operatorname{rank}(\mathbf{C}W - \mathbf{C}W') > 0$$
$$\operatorname{rank}(W - W') > 1 \ \Rightarrow \ \operatorname{rank}(\mathbf{C}W - \mathbf{C}W') > 1$$

Hence by Lemma 14 it holds:

$$\mathbb{P}\big[|\mathbf{V}_i| \leq q \ \vee \ \underbrace{\exists \, (r, S) \in \mathcal{M} \, \forall \, \tilde{v} \in \mathbf{V}_i : \ \tau_i(\mathbf{w}_1, \ldots, \mathbf{w}_{i-1}, \tilde{v}) = r\tilde{v} + S}_{= \neg \mathcal{E}'}\big] \ > \ 1 - (1+q)q^{1-k} - 2 \cdot \iota(\mathbf{C}, \mathbf{W}_i)$$

Moreover, for all $\lambda \in \mathbb{N}$ we can estimate:

$$\mathbb{P}\big[\mathbf{w}_i \in \mathbf{V}_i \ \wedge \ |\mathbf{V}_i| \leq \lambda\big] \ \leq \ \lambda q^{1-k} \tag{1}$$

Thus, we can conclude:

$$\mathbb{P}[\mathbf{w}_i \in \mathbf{V}_i \ \wedge \ \mathcal{E}'] \ < \ (1 + 2q)q^{1-k} + 2 \cdot \iota(\mathbf{C}, \mathbf{W}_i) \tag{2}$$

Furthermore, since $\mathbf{v}_i$ is uniformly random over $\mathbf{V}_i$, we have:

$$\mathbb{P}[\mathbf{w}_i \in \mathbf{V}_i \ \wedge \ \mathbf{v}_i = \mathbf{w}_i] \ = \ \sum_{\lambda=1}^{q^k} \mathbb{P}\big[\mathbf{w}_i \in \mathbf{V}_i \ \wedge \ |\mathbf{V}_i| = \lambda\big] \cdot \tfrac{1}{\lambda} \ \leq \ q^{1-k}$$

Hence, for assertion 2 we already have the following estimation:

$$\mathbb{P}\big[\mathbf{w}_i \in \mathbf{V}_i \ \wedge \ \big(\mathbf{v}_i = \mathbf{w}_i \ \vee \ \mathcal{E}'\big)\big] \ < \ 2(1+q)q^{1-k} + 2 \cdot \iota(\mathbf{C}, \mathbf{W}_i)$$

By Corollary 20 we have that $\iota(\mathbf{C}, \mathbf{W}_i) \leq \iota\big(\mathbf{C}, (\mathbf{w}_1, \ldots, \mathbf{w}_{i-1})\big) < \sqrt{\exp\big((i-1)q^{2-k}\big) - 1}$, what already concludes our proof of assertion 2:

$$\mathbb{P}\big[\mathbf{w}_i \in \mathbf{V}_i \ \wedge \ \big(\mathbf{v}_i = \mathbf{w}_i \ \vee \ \mathcal{E}'\big)\big] \ < \ 2\left((1+q)q^{1-k} + \sqrt{\exp\big((i-1)q^{2-k}\big) - 1}\right)$$

Now, note that by construction the following implications always hold true:

$$\begin{array}{lcl}\mathcal{E} & \Rightarrow & \mathbf{w}_i \in \mathbf{V}_i \ \wedge \ \mathbf{v}_i \in \mathbf{V}_i \ \wedge \ \mathbf{v}_i \neq \mathbf{w}_i \\ \mathcal{E} & \Rightarrow & \tau_i(\mathbf{w}_1, \ldots, \mathbf{w}_i) = \mathbf{r}_i \mathbf{w}_i + \mathbf{S}_i \\ \mathcal{E} & \Rightarrow & \tau_i(\mathbf{w}_1, \ldots, \mathbf{w}_{i-1}, \mathbf{v}_i) = \mathbf{r}_i \mathbf{v}_i + \mathbf{S}_i\end{array}$$

Thus, also the following implication does always hold true:

$$\mathcal{E} \wedge \neg\mathcal{E}' \quad \Rightarrow \quad \forall\, \tilde{v} \in \mathbf{V}_i : \tau_i(\mathbf{w}_1, \ldots, \mathbf{w}_{i-1}, \tilde{v}) = \mathbf{r}_i \tilde{v} + \mathbf{S}_i$$

This yields:

$$\mathcal{E} \wedge \neg\mathcal{E}' \wedge \mathcal{E}'' \quad \Rightarrow \quad |\mathbf{V}_i| < q^{3k/4}$$

Putting things together, we can conclude our proof for assertion 6. The above implications yield:

$$\mathbb{P}[\mathcal{E} \wedge \mathcal{E}''] \quad \leq \quad \mathbb{P}[\mathcal{E} \wedge \neg\mathcal{E}' \wedge \mathcal{E}''] + \mathbb{P}[\mathcal{E} \wedge \mathcal{E}'] \quad \leq \quad \mathbb{P}\big[\mathbf{w}_i \in \mathbf{V}_i \ \wedge \ |\mathbf{V}_i| < q^{3k/4}\big] + \mathbb{P}\big[\mathbf{w}_i \in \mathbf{V}_i \ \wedge \ \mathcal{E}'\big]$$

By (1,2) and Corollary 20 it follows:

$$\mathbb{P}[\mathcal{E} \wedge \mathcal{E}''] \quad \leq \quad q^{1-k/4} + (1+2q)q^{1-k} + 2\sqrt{\exp\big((i-1)q^{2-k}\big) - 1} \qquad \square$$

Game $\Gamma_3(\mathbb{F}_q^k, n, l)$

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 $\mathcal{H} := \mathbb{F}_q^k \setminus \{0\}$ and $\mathcal{C} := \mathbb{F}_q^{3k\times 4k}$ and $\mathcal{M} := \mathbb{F}_q^{4k\times(1+k)}$. The player $\mathcal{K}$ is computationally unbounded.

1. For each $i \in \{l, \ldots, n\}$ the player $\mathcal{K}$ specifies some mapping $\tau_i : \mathcal{U}^l \to \mathcal{M} \cup \{\bot\}$.
2. The player $\mathcal{K}$ learns l random vectors $h_1, \ldots, h_l \overset{\text{r}}{\leftarrow} \mathcal{H}$ and a random matrix $C \overset{\text{r}}{\leftarrow} \mathcal{C}$.
3. For $i = 1, \ldots, l-1$:
   (a) The player $\mathcal{K}$ chooses some $x_i \in \mathbb{F}_q$.
   (b) The player $\mathcal{K}$ lerns $z_i \overset{\text{r}}{\leftarrow} \{\tilde{z} \in \mathcal{U} \mid \tilde{z}h_i = x_i\}$.
4. A random bit $d \overset{\text{r}}{\leftarrow} \{0,1\}$ is chosen secretly. If $d = 0$, let $w \overset{\text{r}}{\leftarrow} \{\tilde{z} \in \mathcal{U} \mid \tilde{z}h_i = x_i\}$, else let $w \overset{\text{r}}{\leftarrow} \mathcal{U}$.
5. For $i = l, \ldots, n$:
   (a) The player $\mathcal{K}$ chooses some $\tilde{M}_i \in \mathbb{F}_q^{3k\times k}$.
   (b) Let $M_i := \tau_i(z_1, \ldots, z_{l-1}, w)$. If $M_i \neq \bot$ and $CM_i = \tilde{M}_i$, the player $\mathcal{K}$ learns M_i; else $\mathcal{K}$ receives a special message $(*, i)$ 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 19: The third version of our indistinguishability game. The main difference from $\Gamma_2(\mathbb{F}_q^k, n, l)$ is a more powerful player $\mathcal{K}$ that effectively may choose $u_{l+1}, \ldots, u_n$ right at the start of the game and automatically learns the values $z_1, \ldots, z_{l-1}$.

Lemma 9. *The maximal winning probability in the game $\Gamma_2(\mathbb{F}_q^k, n, l)$ is upper bounded by the maximal winning probability in $\Gamma_3(\mathbb{F}_q^k, n, l)$.*

Proof. The proof is straightforward, as the game $\Gamma_3(\mathbb{F}_q^k, n, l)$ is actually the same as $\Gamma_2(\mathbb{F}_q^k, n, l)$ with a more powerful player $\mathcal{K}$. In particular, we have the following changes from $\Gamma_2(\mathbb{F}_q^k, n, l)$ to $\Gamma_3(\mathbb{F}_q^k, n, l)$:

- We identify tuples $(\tilde{r}, \tilde{S}) \in \mathbb{F}_q^{4k} \times \mathbb{F}_q^{4k\times k}$ with matrices $\tilde{M} \in \mathbb{F}_q^{4k\times(1+k)}$.
- The player $\mathcal{K}$ learns $z_1, \dots, z_{l-1}$, whereby for $i \in \{1, \dots, l-1\}$ the step 4c of $\Gamma_2(\mathbb{F}_q^k, n, l)$ just becomes needless. Hence, the functions $\tau_1, \dots, \tau_{l-1}$ also are not needed any more.
- Instead of picking $u_{l+1}, \dots, u_n$ uniformly at random, the player $\mathcal{K}$ may choose them right at the start of the game. Thereby, the functions $\tau_l, \dots, \tau_n$ effectively have input domain $\mathcal{U}^l$.
- The functions τ_i now directly map to $\mathcal{M} \cup \{\bot\}$, whereas in step 4c of $\Gamma_2(\mathbb{F}_q^k, n, l)$ an element of $\mathcal{M} \cup \{\bot\}$ was deterministically computed from τ_i and its inputs $w_1, \dots, w_i$. □

Game $\Gamma_4(\mathbb{F}_q^k, n, l)$

Parametrized by a finite vector space $\mathbb{F}_q^k$ and some $n, l \in \mathbb{N}_{>0}$ with $l \le n$. Let $\mathcal{U} := \mathbb{F}_q^{1\times k}$ and $\mathcal{H} := \mathbb{F}_q^k \setminus \{0\}$ and $\mathcal{C} := \mathbb{F}_q^{3k\times 4k}$ and $\mathcal{M} := \mathbb{F}_q^{4k\times(1+k)}$. The player $\mathcal{K}$ is computationally unbounded.

1. For each $i \in \{l, \dots, n\}$ the player $\mathcal{K}$ specifies some mapping $\tau_i : \mathcal{U}^l \to \mathcal{M} \cup \{\bot\}$.
2. The player $\mathcal{K}$ learns l random vectors $h_1, \dots, h_l \xleftarrow{r} \mathcal{H}$.
3. For $i = 1, \dots, l-1$:
   (a) The player $\mathcal{K}$ chooses some $x_i \in \mathbb{F}_q$.
   (b) The player $\mathcal{K}$ lerns $z_i \xleftarrow{r} \{\tilde{z} \in \mathcal{U} \mid \tilde{z}h_i = x_i\}$.
4. A random bit $d \xleftarrow{r} \{0,1\}$ is chosen secretly. If $d = 0$, let $w \xleftarrow{r} \{\tilde{z} \in \mathcal{U} \mid \tilde{z}h_i = x_i\}$, else let $w \xleftarrow{r} \mathcal{U}$.
5. For $i = l, \dots, n$:
   (a) The player $\mathcal{K}$ chooses some $M_i \in \mathcal{M}$.
   (b) If $M_i = \tau_i(z_1, \dots, z_{l-1}, w)$, the player $\mathcal{K}$ is notified about that by the message $(\top, i)$; else $\mathcal{K}$ receives a special message $(*, i)$ 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 20: The fourth version of our indistinguishability game. The only difference from $\Gamma_3(\mathbb{F}_q^k, n, l)$ is that the player $\mathcal{K}$ in step 5 now has to precisely forecast $\tau_i(z_1, \dots, z_{l-1}, w)$ rather than only $C \cdot \tau_i(z_1, \dots, z_{l-1}, w)$.

Lemma 10. *Let $\frac{1}{2} + \delta$ be the maximal winning probability in the game $\Gamma_4(\mathbb{F}_q^k, n, l)$. Then the maximal winning probability in $\Gamma_3(\mathbb{F}_q^k, n, l)$ is upper bounded by:*

$$\tfrac{1}{2} + \delta + (n-l+1)\left(q^{1-k} + \sqrt{\exp\big((l-1)q^{2-k}\big) - 1}\right)$$

Proof. Let the random variables $\mathbf{C} \in \mathcal{C}$ and $\mathbf{z}_1, \ldots, \mathbf{z}_{l-1}, \mathbf{w} \in \mathcal{U}$ represent the same-named random values in the game $\Gamma_3(\mathbb{F}_q^k, n, l)$. W.l.o.g. we assume a deterministic player $\mathcal{K}$, i.e. the mappings $\tau_l, \ldots, \tau_n$ are fixed. Further, for $i = l, \ldots, n$ we set $\mathbf{W}_i := \tau_i(\mathbf{z}_1, \ldots, \mathbf{z}_{l-1}, \mathcal{U})$. Note that the only difference between the games $\Gamma_3(\mathbb{F}_q^k, n, l)$ and $\Gamma_4(\mathbb{F}_q^k, n, l)$ is that in $\Gamma_4(\mathbb{F}_q^k, n, l)$ the player $\mathcal{K}$ would have to precisely forecast the values $\tau_i(\mathbf{z}_1, \ldots, \mathbf{z}_{l-1}, \mathbf{w})$, whereas in $\Gamma_3(\mathbb{F}_q^k, n, l)$ he has only to forecast $\mathbf{C} \cdot \tau_i(\mathbf{z}_1, \ldots, \mathbf{z}_{l-1}, \mathbf{w})$. However, both forecasts are equivalent, whenever $\mathbf{C}$ operates injectively on $\tau_i(\mathbf{z}_1, \ldots, \mathbf{z}_{l-1}, \mathcal{U})$—note that $\mathcal{K}$ always knows $\mathbf{C}$ and $\tau_i(\mathbf{z}_1, \ldots, \mathbf{z}_{l-1}, \mathcal{U})$. In other words, for our proof we just have to upper bound the probability that $\mathbf{C}W = \mathbf{C}W'$ for any distinct $W, W' \in \mathbf{W}_i$. By Lemma 13 and the definition of ι (see Section 2.1) we can estimate for all $i \in \{l, \ldots, n\}$:

$$\mathbb{P}[\exists\, W, W' \in \mathbf{W}_i : \ W \neq W' \wedge \mathbf{C}W = \mathbf{C}W'] \quad < \quad q^{1-3k}|\mathcal{U}|^2 + \iota(\mathbf{C}, \mathbf{W}_i) \quad = \quad q^{1-k} + \iota(\mathbf{C}, \mathbf{W}_i)$$

Moreover, by Corollary 20 we have that $\iota(\mathbf{C}, \mathbf{W}_i) \le \iota\big(\mathbf{C}, (\mathbf{z}_1, \ldots, \mathbf{z}_{l-1})\big) < \sqrt{\exp\big((l-1)q^{2-k}\big) - 1}$ for all $i \in \{l, \ldots, n\}$. Our lemma now follows by the Union Bound. $\square$

Proposition 11. *Let $\frac{1}{2} + \delta$ be the maximal winning probability in the game $\Gamma_4(\mathbb{F}_q^k, n, l)$. Then it holds:*

$$\delta \ < \ (n - l + 1)\left(3q^{(4-k)/3} + q \cdot \sqrt{\exp\big((l-1)q^{2-k}\big) - 1}\right)$$

Proof. Let any player $\mathcal{K}$ for the game $\Gamma_4(\mathbb{F}_q^k, n, l)$ be given, that w.l.o.g. plays deterministically, i.e. the mappings $\tau_l, \ldots, \tau_n$ are fixed. Let the random variables $\mathbf{h}_1, \ldots, \mathbf{h}_l \in \mathcal{H}$, $\mathbf{z}_1, \ldots, \mathbf{z}_{l-1}, \mathbf{w} \in \mathcal{U}$ and $\mathbf{d}, \tilde{\mathbf{d}} \in \{0, 1\}$ represent the same-named random values in the game $\Gamma_4(\mathbb{F}_q^k, n, l)$ with player $\mathcal{K}$, i.e. in particular $\mathbf{h}_1, \ldots, \mathbf{h}_l \xleftarrow{\text{r}} \mathcal{H}$ and $\mathbf{d} \xleftarrow{\text{r}} \{0, 1\}$. Since w.l.o.g. $\mathcal{K}$ plays deterministically, for $i = 1, \ldots, l$ we can write his choice of x_i as a function value $x_i(\mathbf{z}_1, \ldots, \mathbf{z}_{i-1}, \mathbf{h}_1, \ldots, \mathbf{h}_l)$, whereby we get:

$$\mathbf{z}_i \xleftarrow{\text{r}} \big\{\tilde{z} \in \mathcal{U} \ \big|\ \tilde{z}\mathbf{h}_i = x_i(\mathbf{z}_1, \ldots, \mathbf{z}_{i-1}, \mathbf{h}_1, \ldots, \mathbf{h}_l)\big\}$$

Note that thereby the random variables $\mathbf{z}_1, \ldots, \mathbf{z}_{l-1}$ are well-defined. Further, for $i = l, \ldots, n$ let the mapping $\sigma_i : \mathcal{U}^{l-1} \times \mathcal{H}^l \to \mathcal{M}$ represent how the player $\mathcal{K}$ computes M_i from $(\mathbf{z}_1, \ldots, \mathbf{z}_{l-1}, \mathbf{h}_1, \ldots, \mathbf{h}_l)$. Finally, 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. $\sigma_i(\mathbf{z}_1, \ldots, \mathbf{z}_{l-1}, \mathbf{h}_1, \ldots, \mathbf{h}_l) = \tau_i(\mathbf{z}_1, \ldots, \mathbf{z}_{l-1}, \mathbf{w})$ for all $i \in \{l, \ldots, \mathbf{m}\}$ and $\sigma_{\mathbf{m}+1}(\mathbf{z}_1, \ldots, \mathbf{z}_{l-1}, \mathbf{h}_1, \ldots, \mathbf{h}_l) \neq \tau_{\mathbf{m}+1}(\mathbf{z}_1, \ldots, \mathbf{z}_{l-1}, \mathbf{w})$. Note that the view of $\mathcal{K}$ can be completely reconstructed from $(\mathbf{z}_1, \ldots, \mathbf{z}_{l-1}, \mathbf{h}_1, \ldots, \mathbf{h}_l, \mathbf{m})$ and $\mathcal{K}$'s program code. To make formulas a bit shorter, we set $\mathbf{R} := (\mathbf{z}_1, \ldots, \mathbf{z}_{l-1}, \mathbf{h}_1, \ldots, \mathbf{h}_{l-1})$. Let $\mathcal{R}$ denote the support of $\mathbf{R}$. Thus we get:

$$\begin{aligned}
&\mathbb{P}[\tilde{\mathbf{d}} = \mathbf{d}] \\
&= \mathbb{P}[\tilde{\mathbf{d}} = 0 \mid \mathbf{d} = 0] \cdot \mathbb{P}[\mathbf{d} = 0] + \mathbb{P}[\tilde{\mathbf{d}} = 1 \mid \mathbf{d} = 1] \cdot \mathbb{P}[\mathbf{d} = 1] \\
&= \tfrac{1}{2}\Big(\mathbb{P}[\tilde{\mathbf{d}} = 0 \mid \mathbf{d} = 0] + \mathbb{P}[\tilde{\mathbf{d}} = 1 \mid \mathbf{d} = 1]\Big) \\
&= \tfrac{1}{2}\Big(\mathbb{P}[\tilde{\mathbf{d}} = 0 \mid \mathbf{d} = 0] + 1 - \mathbb{P}[\tilde{\mathbf{d}} = 0 \mid \mathbf{d} = 1]\Big) \\
&\le \tfrac{1}{2} + \tfrac{1}{2}\Big|\mathbb{P}[\tilde{\mathbf{d}} = 0 \mid \mathbf{d} = 0] - \mathbb{P}[\tilde{\mathbf{d}} = 0 \mid \mathbf{d} = 1]\Big| \\
&\le \tfrac{1}{2} + \tfrac{1}{2}\sum_{\substack{R \in \mathcal{R} \\ h \in \mathcal{H}}} \sum_{m=l-1}^{n} \Big|\mathbb{P}\big[(\mathbf{R}, \mathbf{h}_l, \mathbf{m}) = (R, h, m) \ \big|\ \mathbf{d} = 0\big] - \mathbb{P}\big[(\mathbf{R}, \mathbf{h}_l, \mathbf{m}) = (R, h, m) \ \big|\ \mathbf{d} = 1\big]\Big| \qquad (3)
\end{aligned}$$

The last inequality is by the fact that the value of $\tilde{\mathbf{d}}$ can be deterministically computed from $(\mathbf{z}_1,\dots,\mathbf{z}_{l-1},\mathbf{h}_1,\dots,\mathbf{h}_l,\mathbf{m})$, as mentioned above. Now, for $m = l-1,\dots,n+1$ and $(R,h) \in \mathcal{R}\times\mathcal{H}$ we define the following sets:

$$\bar{A}_m(R,h) := \big\{\tilde{v} \in \mathcal{U} \bigm| (\mathbf{R},\mathbf{h}_l,\mathbf{w}) = (R,h,\tilde{v}) \;\Rightarrow\; \mathbf{m} = m\big\}$$
$$A_m(R,h) := \big\{\tilde{v} \in \mathcal{U} \bigm| (\mathbf{R},\mathbf{h}_l,\mathbf{w}) = (R,h,\tilde{v}) \;\Rightarrow\; \mathbf{m} \geq m\big\}$$

I.e., $A_m(\mathbf{R},\mathbf{h}_l)$ is exactly the set of vectors $\tilde{v} \in \mathcal{U}$, such that $\tau_l(\mathbf{z}_1,\dots,\mathbf{z}_{l-1},\tilde{v}),\dots,\tau_m(\mathbf{z}_1,\dots,\mathbf{z}_{l-1},\tilde{v})$ match $\mathcal{K}$'s forecasts $\sigma_l(\mathbf{z}_1,\dots,\mathbf{z}_{l-1},\mathbf{h}_1,\dots,\mathbf{h}_l),\dots,\sigma_m(\mathbf{z}_1,\dots,\mathbf{z}_{l-1},\mathbf{h}_1,\dots,\mathbf{h}_l)$. Finally, for $\alpha \in \mathbb{F}_q$ and $h \in \mathcal{H}$ let $Z_\alpha(h) := \big\{\tilde{v} \in \mathcal{V} \bigm| \tilde{v}h = \alpha\big\}$, i.e. we have that $\mathbf{z}_i \overset{\mathrm{r}}{\leftarrow} Z_{x_i(\mathbf{z}_1,\dots,\mathbf{z}_{i-1},\mathbf{h}_1,\dots,\mathbf{h}_l)}(\mathbf{h}_i)$. Note that $\mathbf{w} \overset{\mathrm{r}}{\leftarrow} Z_{x_l(\mathbf{R},\mathbf{h}_l)}$ if $\mathbf{d} = 0$, and that $\mathbf{w} \overset{\mathrm{r}}{\leftarrow} \mathcal{U}$ if $\mathbf{d} = 1$. Hence, given any $R \in \mathcal{R}$ and $h \in \mathcal{H}$, we can compute:

$$\begin{aligned}
&\sum_{m=l-1}^{n} \Big|\mathbb{P}\big[\mathbf{m} = m \bigm| (\mathbf{d},\mathbf{R},\mathbf{h}_l) = (0,R,h)\big] - \mathbb{P}\big[\mathbf{m} = m \bigm| (\mathbf{d},\mathbf{R},\mathbf{h}_l) = (1,R,h)\big]\Big| \\
&= \sum_{m=l-1}^{n} \Big|\mathbb{P}\big[\mathbf{w} \in \bar{A}_m(R,h) \bigm| (\mathbf{d},\mathbf{R},\mathbf{h}_l) = (0,R,h)\big] - \mathbb{P}\big[\mathbf{w} \in \bar{A}_m(R,h) \bigm| (\mathbf{d},\mathbf{R},\mathbf{h}_l) = (1,R,h)\big]\Big| \\
&= \sum_{m=l-1}^{n} \left| \frac{\big|Z_{x_l(R,h)}(h) \cap \bar{A}_m(R,h)\big|}{\big|Z_{x_l(R,h)}(h)\big|} - \frac{\big|\bar{A}_m(R,h)\big|}{|\mathcal{U}|} \right| \\
&= q^{1-k} \sum_{m=l-1}^{n} \Big|\big|Z_{x_l(R,h)}(h) \cap \bar{A}_m(R,h)\big| - \tfrac{1}{q}\big|\bar{A}_m(R,h)\big|\Big|
\end{aligned}$$

Putting this together with (3), we can conclude:

$$\mathbb{P}[\tilde{\mathbf{d}} = \mathbf{d}] \;\leq\; \tfrac{1}{2} + \tfrac{q^{1-k}}{2} \sum_{m=l-1}^{n} \mathbb{E}\Big|\big|Z_{x_l(\mathbf{R},\mathbf{h}_l)}(\mathbf{h}_l) \cap \bar{A}_m(\mathbf{R},\mathbf{h}_l)\big| - \tfrac{1}{q}\big|\bar{A}_m(\mathbf{R},\mathbf{h}_l)\big|\Big|$$

However, by construction we have that $\big|V \cap \bar{A}_m(R,h)\big| = \big|V \cap A_m(R,h)\big| - \big|V \cap A_{m+1}(R,h)\big|$ for all $V \subseteq \mathcal{U}$, since $\bar{A}_m(R,h) = A_m(R,h) \setminus A_{m+1}(R,h)$ and $A_m(R,h) \supseteq A_{m+1}(R,h)$. Also note that $A_{l-1}(R,h) = \mathcal{U}$ and $A_{n+1}(R,h) = \emptyset$. Using this and the Triangle Inequality, we can derive:

$$\mathbb{P}[\tilde{\mathbf{d}} = \mathbf{d}] \;<\; \tfrac{1}{2} + q^{1-k} \sum_{m=l}^{n} \mathbb{E}\Big|\big|Z_{x_l(\mathbf{R},\mathbf{h}_l)}(\mathbf{h}_l) \cap A_m(\mathbf{R},\mathbf{h}_l)\big| - \tfrac{1}{q}\big|A_m(\mathbf{R},\mathbf{h}_l)\big|\Big| \tag{4}$$

Now let $\mathbf{t} := (\mathbf{z}_1,\dots,\mathbf{z}_{l-1})$. For $m = l,\dots,n$ and $H \in \mathcal{H}^l$ we can rewrite $A_m(\mathbf{t},H)$ as follows:

$$A_m(\mathbf{t},H) \;=\; \big\{\tilde{v} \in \mathcal{U} \bigm| \forall j \in \{l,\dots,m\} : \tau_j(\mathbf{t},\tilde{v}) = \sigma_j(\mathbf{t},H)\big\}$$

Thus, we always have for all $H, H' \in \mathcal{H}^l$ and $m = l,\dots,n$ that $A_m(\mathbf{t},H) \cap A_m(\mathbf{t},H') = \emptyset$ or $A_m(\mathbf{t},H) = A_m(\mathbf{t},H')$. By Corollary 17 for arbitrary $\gamma \in \mathbb{R}_{>0}$ follows:

$$\mathbb{P}\Big[\exists\, \alpha \in \mathbb{F}_q,\, H \in \mathcal{H}^l : \Big|\big|Z_\alpha(\mathbf{h}_l) \cap A_m(\mathbf{t},H)\big| - \tfrac{1}{q}\big|A_m(\mathbf{t},H)\big|\Big| > \gamma\Big] \;\leq\; \frac{q^{k+1/2}}{\gamma^{3/2}} + \iota(\mathbf{h}_l,\mathbf{t})$$

Thus, for $m = l, \ldots, n$ and arbitrary $\gamma \in \mathbb{R}_{>0}$ we especially have:

$$\mathbb{P}\left[\left|\left|Z_{x_l(\mathbf{R},\mathbf{h}_l)}(\mathbf{h}_l) \cap A_m(\mathbf{R},\mathbf{h}_l)\right| - \tfrac{1}{q}\left|A_m(\mathbf{R},\mathbf{h}_l)\right|\right| > \gamma\right] \;\leq\; \frac{q^{k+1/2}}{\gamma^{3/2}} + \iota(\mathbf{h}_l, \mathbf{t})$$

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 yields:

$$\begin{aligned}\mathbb{E}\left|\left|Z_{x_l(\mathbf{R},\mathbf{h}_l)}(\mathbf{h}_l) \cap A_m(\mathbf{R},\mathbf{h}_l)\right| - \tfrac{1}{q}\left|A_m(\mathbf{R},\mathbf{h}_l)\right|\right| \;&\leq\; \int_0^{q^k} \min\left\{1, \tfrac{q^{k+1/2}}{\gamma^{3/2}}\right\} + \iota(\mathbf{h}_l, \mathbf{t})\, \mathrm{d}\gamma \\ &<\; 3q^{(2k+1)/3} + q^k \cdot \iota(\mathbf{h}_l, \mathbf{t})\end{aligned}$$

Moreover, by Corollary 20 we have that $\iota(\mathbf{h}_l, \mathbf{t}) < \sqrt{\exp\left((l-1)q^{2-k}\right) - 1}$. Using (4), we conclude:

$$\begin{aligned}\mathbb{P}[\tilde{\mathbf{d}} = \mathbf{d}] \;&<\; \tfrac{1}{2} + q^{1-k}(n-l+1)\left(3q^{(2k+1)/3} + q^k \cdot \sqrt{\exp\left((l-1)q^{2-k}\right) - 1}\right) \\ &=\; \tfrac{1}{2} + (n-l+1)\left(3q^{(4-k)/3} + q \cdot \sqrt{\exp\left((l-1)q^{2-k}\right) - 1}\right)\end{aligned}$$

□

Corollary 12. *The statistical distance between the environment's view in the hybrid game* Game_0 *and its view in* Game_n *is negligible in the security parameter* $k \log q$*, if only* $k \geq 5$*.*

Proof. For $i = 1, \ldots, 4$ let δ_i denote the maximal advantage of the player $\mathcal{K}$ in the game $\Gamma_i(\mathbb{F}_q^k, n, l)$, i.e. the respective winning probability is $\frac{1}{2} + \delta_i$. By Lemma 7 we know that the statistical distance between the environment's view in the hybrid game Game_{l-1} and its view in Game_l is upper bounded by $2\delta_1$. Furthermore, it holds:

$$\begin{aligned}
\delta_1 \;&\leq\; \delta_2 + n\left(2q^{1-k/4} + 3(1+2q)q^{1-k}\right) + 4\sum\nolimits_{i=0}^{n-1} \sqrt{\exp(iq^{2-k}) - 1} && \text{by Lemma 8, if } q^k \geq 22 \\
\delta_2 \;&\leq\; \delta_3 && \text{by Lemma 9} \\
\delta_3 \;&\leq\; \delta_4 + (n-l+1)\left(q^{1-k} + \sqrt{\exp\left((l-1)q^{2-k}\right) - 1}\right) && \text{by Lemma 10} \\
\delta_4 \;&<\; (n-l+1)\left(3q^{(4-k)/3} + q \cdot \sqrt{\exp\left((l-1)q^{2-k}\right) - 1}\right) && \text{by Proposition 11}
\end{aligned}$$

Thus, exploiting that $1 \leq l \leq n$ we can upper bound the statistical distance between the environment's views in Game_{l-1} and Game_l by:

$$2n\left((6q+4)q^{1-k} + 3q^{(4-k)/3} + 2q^{1-k/4} + (q+5)\sqrt{\exp(nq^{2-k}) - 1}\right) \tag{5}$$

Note that the last three summands within the brackets are negligible in $k \log q$ if only $k \geq 5$, and the first summand is negligible if only $k \geq 3$. Since the statistical distance between the environment's views in Game_0 and Game_n is at most by a factor n bigger than (5), this concludes our proof. □

4.3.3 Technical lemmata

In this section we develop the technical tools our security proof is based on. The following is a high level overview of what we are going to show:

- Our first technical lemma (Lemma 13) yields an upper bound for the probability that in our protocol $\Pi_{\text{OAFE}}^{\text{semi-int}}$ the random matrix C cancels differences between token outputs in the sense that $\operatorname{rank}\bigl(\tau_i(z)-\tau_i(z')\bigr) > \operatorname{rank}\bigl(C\cdot\bigl(\tau_i(z)-\tau_i(z')\bigr)\bigr)$ with τ_i denoting the token functionality in stage i.
- Next (in Lemma 14), we exploit these rank preserving properties of C to show that the token functionality τ is linear on *all* inputs that do not cause a protocol abortion.
- In Section 3.1 we argued that the randomly chosen h_i partition token input sets into subsets of roughly equal size. Using an appropriate version of the well-known Leftover Hash Lemma (Lemma 15) as a technical tool, we establish an upper bound for the variance of the cardinality of these parts (Lemma 16 and Corollary 17). The main technical difficulty herein is to take into account that the token's current behavior may depend on all previous inputs, which are absolutely *not* independent of the environment's view.
- By Lemma 18 we estimate the number of token inputs belonging to two or more different linear functions that a simulator is likely to extract. We show that these inputs are only a negligible fraction of all possible token inputs.
- Finally, we show that the token inputs $z_1,\dots z_n$ are indistinguishable from pure randomness (Lemma 19) and thus the correlation between the token's view and David's inputs $x_1,\dots,x_n$ is negligible (Corollary 20).

Lemma 13. *Let $\mathbb{F}_q$ be some finite field of size $q\geq 2$ and let $l,m,n,r\in\mathbb{N}_{>0}$ with $r\leq\min(l,m,n)$. Then for arbitrary $\mathcal{W}\subseteq\mathbb{F}_q^{m\times n}$ and $\mathbf{C}\xleftarrow{\text{r}}\mathbb{F}_q^{l\times m}$ it does hold:*

$$\mathbb{P}\bigl[\exists\, W\in\mathcal{W}:\ \operatorname{rank}(W)\geq r>\operatorname{rank}(\mathbf{C}W)\bigr]\ <\ q^{r-l}|\mathcal{W}|$$

Proof. We first estimate the number of matrices in $\mathbb{F}_q^{l\times r}$ that have full rank r. Given $i\in\{1,\dots,r\}$ and any matrix $C\in\mathbb{F}_q^{l\times i}$ with $\operatorname{rank}(C)=i$, 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:

$$\#\bigl\{C\in\mathbb{F}_q^{l\times r}\ \big|\ \operatorname{rank}(C)=r\bigr\}\ =\ \prod_{i=0}^{r-1}q^l-q^i$$

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 $W\in\mathbb{F}_q^{m\times n}$ be some arbitrary matrix with $\bar{r}:=\operatorname{rank}(W)\geq r$. Further let $\bar{B}\in\mathbb{F}_q^{\bar{r}\times n}$, such that $\bar{B}$ only consists of linearly independent rows of W, i.e. especially $\bar{B}$ has full rank $\bar{r}$. Let

$B \in \mathbb{F}_q^{m\times n}$, 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:

$$\begin{aligned}
\#\big\{C \in \mathbb{F}_q^{l\times m} \;\big|\; \mathrm{rank}(CW) < r\big\} &= q^{lm} - \#\big\{C \in \mathbb{F}_q^{l\times m} \;\big|\; \mathrm{rank}(CW) \geq r\big\} \\
&= q^{lm} - \#\big\{C \in \mathbb{F}_q^{l\times m} \;\big|\; \mathrm{rank}(CMB) \geq r\big\} \\
&= q^{lm} - \#\big\{C \in \mathbb{F}_q^{l\times m} \;\big|\; \mathrm{rank}(CB) \geq r\big\} \\
&= q^{lm} - \#\big\{C \in \mathbb{F}_q^{l\times \bar{r}} \;\big|\; \mathrm{rank}(C\bar{B}) \geq r\big\} \cdot q^{l(m-\bar{r})} \\
&= q^{lm} - \#\big\{C \in \mathbb{F}_q^{l\times \bar{r}} \;\big|\; \mathrm{rank}(C) \geq r\big\} \cdot q^{l(m-\bar{r})} \\
&\leq q^{lm} - \#\big\{C \in \mathbb{F}_q^{l\times r} \;\big|\; \mathrm{rank}(C) = r\big\} \cdot q^{l(m-r)} \\
&< q^{lm} - q^{lm}\Big(1 - q^{r-l}\Big) \\
&= q^{l(m-1)+r}
\end{aligned}$$

Thereby, for arbitrary $W \in \mathbb{F}_q^{m\times n}$ and $\mathbf{C} \overset{\mathrm{r}}{\leftarrow} \mathbb{F}_q^{l\times m}$ we can conclude:

$$\mathbb{P}\big[\mathrm{rank}(W) \geq r > \mathrm{rank}(\mathbf{C}W)\big] \;<\; q^{r-l} \qquad \square$$

Lemma 14. *Let $\mathbb{F}_q$ be some finite field of size $q \geq 2$ and let $l, m, n \in \mathbb{N}_{>0}$. Let $\tau : \mathbb{F}_q^{1\times n} \to \mathbb{F}_q^{m\times n}$ be some arbitrary mapping and let $C \in \mathbb{F}_q^{l\times m}$, such that for all $v, v' \in \mathbb{F}_q^{1\times n}$ the following implications hold true:*

$$\begin{aligned}
\mathrm{rank}\big(\tau(v) - \tau(v')\big) > 0 \;&\Rightarrow\; \mathrm{rank}\big(C\cdot\tau(v) - C\cdot\tau(v')\big) > 0 \\
\mathrm{rank}\big(\tau(v) - \tau(v')\big) > 1 \;&\Rightarrow\; \mathrm{rank}\big(C\cdot\tau(v) - C\cdot\tau(v')\big) > 1
\end{aligned}$$

Moreover, let $\tilde{r} \in \mathbb{F}_q^l$, $\tilde{S} \in \mathbb{F}_q^{l\times n}$ and $V := \big\{v \in \mathbb{F}_q^{1\times n} \;\big|\; C\cdot\tau(v) = \tilde{r}\cdot v + \tilde{S}\big\}$, such that $|V| > q$. Then there exist some $r \in \mathbb{F}_q^m$, $S \in \mathbb{F}_q^{m\times n}$, such that $\tau(v) = r\cdot v + S$ for all $v \in V$.

Proof. If $\tilde{r} = 0$ the proof is trivial, since in this case by our choice of C and V we have that $\tau(v) = S$ for all $v \in V$ and some constant $S \in \mathbb{F}_q^{m\times n}$. So, w.l.o.g. let $\tilde{r} \neq 0$.

First of all, we now observe for all $v, v' \in V$ that $\mathrm{rank}(\tau(v) - \tau(v')) \leq 1$, since else by our choice of C we had the contradiction that $1 < \mathrm{rank}\big(C\cdot\tau(v) - C\cdot\tau(v')\big) = \mathrm{rank}\big(\tilde{r}\cdot(v - v')\big) \leq 1$. Thereby, for each $v, v' \in V$ we find some $r \in \mathbb{F}_q^m$ and $\bar{v} \in \mathbb{F}_q^{1\times n}$, 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:

$$\operatorname{rank}\big(r \cdot (v - v') + r' \cdot (v' - v'')\big) \quad = \quad \operatorname{rank}\big(\tau(v) - \tau(v'')\big) \quad \leq \quad 1$$

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, finally we can 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:

$$\tau(\tilde{v}') \quad = \quad \big(\tau(\tilde{v}') - \tau(\tilde{v})\big) + \tau(\tilde{v}) \quad = \quad \big(r \cdot (\tilde{v}' - \tilde{v})\big) + \big(r \cdot \tilde{v} + S\big) \quad = \quad r \cdot \tilde{v}' + S \qquad \square$$

Lemma 15 (Leftover Hash Lemma [BBR88, ILL89])**.** *Let $\mathcal{G}$ be a 2-universal class of functions $\mathcal{X} \to \mathcal{Y}$ and let $\mathbf{g} \xleftarrow{\mathrm{r}} \mathcal{G}$, 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 $\big(\mathbf{g}(\mathbf{x}), \mathbf{g}\big)$ and uniform randomness $(\mathbf{u}, \mathbf{g})$, i.e. $\mathbf{u} \xleftarrow{\mathrm{r}} \mathcal{Y}$, it holds:*

$$\Delta\big((\mathbf{g}(\mathbf{x}), \mathbf{g}), (\mathbf{u}, \mathbf{g})\big) \ \leq \ \tfrac{1}{2}\sqrt{2^{\log_2 |\mathcal{Y}| - \mathbb{H}_2(\mathbf{x})}}$$

Proof. We adapt the proof from [AB09]. Let $(\mathbf{g}', \mathbf{x}')$ be identically distributed as its unprimed counterpart $(\mathbf{g}, \mathbf{x})$. Thereby, when we treat the distribution of $\big(\mathbf{g}(\mathbf{x}), \mathbf{g}\big)$ as a probability vector $\vec{p} \in \mathbb{R}^{\mathcal{Y} \times \mathcal{G}}$, we get:

$$\begin{aligned}
\|\vec{p}\|_2^2 &= \mathbb{P}\big[(\mathbf{g}(\mathbf{x}), \mathbf{g}) = (\mathbf{g}'(\mathbf{x}'), \mathbf{g}')\big] \\
&= \mathbb{P}[\mathbf{g} = \mathbf{g}'] \cdot \mathbb{P}\big[\mathbf{g}(\mathbf{x}) = \mathbf{g}'(\mathbf{x}') \bigm| \mathbf{g} = \mathbf{g}'\big] \\
&= \mathbb{P}[\mathbf{g} = \mathbf{g}'] \cdot \mathbb{P}\big[\mathbf{g}(\mathbf{x}) = \mathbf{g}(\mathbf{x}')\big] \\
&= \mathbb{P}[\mathbf{g} = \mathbf{g}'] \cdot \big(\mathbb{P}[\mathbf{x} = \mathbf{x}'] + \mathbb{P}[\mathbf{x} \neq \mathbf{x}'] \cdot \mathbb{P}\big[\mathbf{g}(\mathbf{x}) = \mathbf{g}(\mathbf{x}') \bigm| \mathbf{x} \neq \mathbf{x}'\big]\big) \\
&\leq \mathbb{P}[\mathbf{g} = \mathbf{g}'] \cdot \big(\mathbb{P}[\mathbf{x} = \mathbf{x}'] + \mathbb{P}\big[\mathbf{g}(\mathbf{x}) = \mathbf{g}(\mathbf{x}') \bigm| \mathbf{x} \neq \mathbf{x}'\big]\big) \\
&= |\mathcal{G}|^{-1} \cdot \Big(2^{-\mathbb{H}_2(\mathbf{x})} + \mathbb{P}\big[\mathbf{g}(\mathbf{x}) = \mathbf{g}(\mathbf{x}') \bigm| \mathbf{x} \neq \mathbf{x}'\big]\Big) \\
&\leq |\mathcal{G}|^{-1} \cdot \Big(2^{-\mathbb{H}_2(\mathbf{x})} + |\mathcal{Y}|^{-1}\Big)
\end{aligned}$$

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:

$$\|\vec{p} - \vec{u}\|_2^2 \quad = \quad \|\vec{p}\|_2^2 - \|\vec{u}\|_2^2 \quad \leq \quad |\mathcal{G}|^{-1} \cdot \Big(2^{-\mathbb{H}_2(\mathbf{x})} + |\mathcal{Y}|^{-1}\Big) - |\mathcal{Y} \times \mathcal{G}|^{-1} \quad = \quad |\mathcal{G}|^{-1} \cdot 2^{-\mathbb{H}_2(\mathbf{x})}$$

Finally, since $\|\vec{v}\|_1 \leq \sqrt{|\mathcal{Y} \times \mathcal{G}|} \cdot \|\vec{v}\|_2$ for every $\vec{v} \in \mathbb{R}^{\mathcal{Y} \times \mathcal{G}}$, we can conclude:

$$\Delta\big((\mathbf{g}(\mathbf{x}), \mathbf{g}), (\mathbf{u}, \mathbf{g})\big) \ = \ \tfrac{1}{2}\|\vec{p} - \vec{u}\|_1 \ \leq \ \tfrac{1}{2}\sqrt{|\mathcal{Y} \times \mathcal{G}|} \cdot \|\vec{p} - \vec{u}\|_2 \ \leq \ \tfrac{1}{2}\sqrt{2^{\log_2 |\mathcal{Y}| - \mathbb{H}_2(\mathbf{x})}} \qquad \square$$

Lemma 16. *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 $Z_\alpha(h) := \{z \in \mathbb{F}_q^{1\times k} \mid z \cdot h = \alpha\}$. Further let $A \subseteq \mathbb{F}_q^{1\times k}$ and $\mathcal{H} := \mathbb{F}_q^k \setminus \{0\}$. Finally, let $x : \mathcal{H} \to \mathbb{F}_q$ be some arbitrary function and let $\mathbf{h} \xleftarrow{\mathrm{r}} \mathcal{H}$. Then 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. Let $\mathbf{a} \xleftarrow{\mathrm{r}} A$ and $\mathbf{u} \xleftarrow{\mathrm{r}} \mathbb{F}_q$. On the one hand, since for all distinct $a, a' \in A$ it holds that $\mathbb{P}[a\mathbf{h} = a'\mathbf{h}] = \mathbb{P}[(a - a')\mathbf{h} = 0] = \frac{q^{k-1}-1}{q^k} < \frac{1}{|\mathbb{F}_q|}$, we can estimate the statistical distance $\delta := \Delta\big((\mathbf{ah}, \mathbf{h}), (\mathbf{u}, \mathbf{h})\big)$ by the Leftover Hash Lemma (Lemma 15) as follows:

$$\delta \;\leq\; \tfrac{1}{2}\sqrt{2^{\log_2|\mathbb{F}_q| - \mathbb{H}_2(\mathbf{a})}} \;=\; \tfrac{1}{2}\sqrt{\tfrac{q}{|A|}}$$

On the other hand, we can estimate:

$$\begin{aligned}
\delta &= \tfrac{1}{2}\textstyle\sum_{\alpha\in\mathbb{F}_q,\, h\in\mathcal{H}} \big|\mathbb{P}[\mathbf{ah} = \alpha \;\wedge\; \mathbf{h} = h] - \mathbb{P}[\mathbf{u} = \alpha \;\wedge\; \mathbf{h} = h]\big| \\
&= \tfrac{1}{2}\textstyle\sum_{\alpha\in\mathbb{F}_q,\, h\in\mathcal{H}} \mathbb{P}[\mathbf{h} = h] \cdot \big|\mathbb{P}[\mathbf{a}h = \alpha] - \mathbb{P}[\mathbf{u} = \alpha]\big| \\
&= \tfrac{1}{2}\textstyle\sum_{\alpha\in\mathbb{F}_q,\, h\in\mathcal{H}} \mathbb{P}[\mathbf{h} = h] \cdot \left|\frac{|A\cap Z_\alpha(h)|}{|A|} - \frac{1}{q}\right| \\
&\geq \tfrac{1}{2}\textstyle\sum_{h\in\mathcal{H}} \mathbb{P}[\mathbf{h} = h] \cdot \left|\frac{|A\cap Z_{x(h)}(h)|}{|A|} - \frac{1}{q}\right| \\
&= \tfrac{1}{2}\,\mathbb{E}\left|\frac{|A\cap Z_{x(\mathbf{h})}(\mathbf{h})|}{|A|} - \frac{1}{q}\right|
\end{aligned}$$

By the linearity of expected values follows:

$$\mathbb{E}\left|\left|A \cap Z_{x(\mathbf{h})}(\mathbf{h})\right| - \tfrac{1}{q}|A|\right| \;\leq\; 2\delta \cdot |A| \;\leq\; \sqrt{q \cdot |A|}$$

□

Corollary 17. *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 $\mathcal{Q}, \mathcal{R}$ be some arbitrary finite sets. Moreover, let some mapping $A : \mathcal{Q} \times \mathcal{R} \to \mathcal{P}(\mathbb{F}_q^{1\times k})$ be given, such that for all $t \in \mathcal{Q}$ and $\nu, \nu' \in \mathcal{R}$ either $A(t,\nu) \cap A(t,\nu') = \emptyset$ or $A(t,\nu) = A(t,\nu')$. 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 $\mathbf{h} \xleftarrow{\mathrm{r}} \mathcal{H}$. Then for every random variable $\mathbf{t} \in \mathcal{Q}$ and arbitrary $\gamma \in \mathbb{R}_{>0}$ it holds:*

$$\mathbb{P}\left[\exists\, \alpha \in \mathbb{F}_q,\, \nu \in \mathcal{R} : \left|\left|A(\mathbf{t},\nu) \cap Z_\alpha(\mathbf{h})\right| - \tfrac{1}{q}\left|A(\mathbf{t},\nu)\right|\right| > \gamma\right] \;\leq\; \frac{q^{k+1/2}}{\gamma^{3/2}} + \iota(\mathbf{h}, \mathbf{t})$$

Proof. It obviously suffices to give a proof for the case that $\mathbf{t}$ and $\mathbf{h}$ are independent, i.e. $\iota(\mathbf{h}, \mathbf{t}) = 0$. Moreover, w.l.o.g. we have that $\mathbb{P}[\mathbf{t} = t] = 1$ for some worst case constant $t \in \mathcal{Q}$. However, once we have fixed $\mathbf{t}$, we can consider α and ν function values of $\mathbf{h}$, which we denote by $\alpha(\mathbf{h})$ and $\nu_{\mathbf{h}}$ respectively. For each $h \in \mathcal{H}$ we 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:

$$\mathbb{P}\left[\left|\left|A(t,\nu_{\mathbf{h}}) \cap Z_{\alpha(\mathbf{h})}(\mathbf{h})\right| - \tfrac{1}{q}\left|A(t,\nu_{\mathbf{h}})\right|\right| > \gamma\right] \;\leq\; \sum_{h\in\bar{\mathcal{H}}} \mathbb{P}\left[\left|\left|A(t,\nu_h) \cap Z_{\alpha(\mathbf{h})}(\mathbf{h})\right| - \tfrac{1}{q}\left|A(t,\nu_h)\right|\right| > \gamma\right]$$

Let $\hat{\mathcal{H}} := \{h \in \bar{\mathcal{H}} \mid \gamma < |A(t,\nu_h)|\}$. Since always $\Big||X \cap Z_{\alpha(\mathbf{h})}(\mathbf{h})| - \frac{1}{q}|X|\Big| < |X|$ for all $X \subseteq \mathbb{F}_q^{1\times k}$, it follows:

$$\sum_{h\in\bar{\mathcal{H}}} \mathbb{P}\Big[\Big||A(t,\nu_h) \cap Z_{\alpha(\mathbf{h})}(\mathbf{h})| - \tfrac{1}{q}|A(t,\nu_h)|\Big| > \gamma\Big] \;=\; \sum_{h\in\hat{\mathcal{H}}} \mathbb{P}\Big[\Big||A(t,\nu_h) \cap Z_{\alpha(\mathbf{h})}(\mathbf{h})| - \tfrac{1}{q}|A(t,\nu_h)|\Big| > \gamma\Big]$$

However, by Lemma 16 we can estimate for all $\nu \in \mathcal{Q}$:

$$\mathbb{P}\Big[\Big||A(t,\nu) \cap Z_{\alpha(\mathbf{h})}(\mathbf{h})| - \tfrac{1}{q}|A(t,\nu)|\Big| \geq \gamma\Big] \;\leq\; \tfrac{1}{\gamma}\sqrt{q \cdot |A(t,\nu)|}$$

Thus, we can conclude:

$$\mathbb{P}\Big[\Big||A(t,\nu_{\mathbf{h}}) \cap Z_{\alpha(\mathbf{h})}(\mathbf{h})| - \tfrac{1}{q}|A(t,\nu_{\mathbf{h}})|\Big| > \gamma\Big] \;\leq\; \tfrac{1}{\gamma}\sum\nolimits_{h\in\hat{\mathcal{H}}} \sqrt{q \cdot |A(t,\nu_h)|}$$

By the standard estimation between the L^1-norm and L^2-norm of $\mathbb{R}$-vectors we have:

$$\sum\nolimits_{h\in\hat{\mathcal{H}}} \sqrt{|A(t,\nu_h)|} \;\leq\; \sqrt{|\hat{\mathcal{H}}| \cdot \sum\nolimits_{h\in\hat{\mathcal{H}}} |A(t,\nu_h)|}$$

Since $\mathbb{F}_q^{1\times k} \supseteq \dot{\bigcup}_{h\in\hat{\mathcal{H}}} A(t,\nu_h)$, we can further estimate:

$$\sum\nolimits_{h\in\hat{\mathcal{H}}} |A(t,\nu_h)| \;\leq\; |\mathbb{F}_q^{1\times k}| \;=\; q^k$$

Note that by construction $|\hat{\mathcal{H}}| < \frac{q^k}{\gamma}$. Putting things together, we have shown:

$$\mathbb{P}\Big[\Big||A(t,\nu_{\mathbf{h}}) \cap Z_{\alpha(\mathbf{h})}(\mathbf{h})| - \tfrac{1}{q}|A(t,\nu_{\mathbf{h}})|\Big| > \gamma\Big] \;<\; \frac{q^{k+1/2}}{\gamma^{3/2}} \qquad \square$$

Lemma 18. *Let $\mathbb{F}_q$ be some finite field of size $q \geq 2$ and let $l, k \in \mathbb{N}_{>0}$, such that $q^k \leq q^{9k/8} - q^{3k/4}$. Let $\tau : \mathbb{F}_q^{1\times k} \to \mathbb{F}_q^{l\times k} \cup \{\bot\}$ 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 properties:*

$$\tau(v) = rv + S \quad \text{and} \quad \#\{\tilde{v} \in \mathbb{F}_q^{1\times k} \mid \tau(\tilde{v}) = r\tilde{v} + S\} \geq q^{3k/4}$$

Then we have that $|V'| < q^{3k/4}$.

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 $r \in \mathbb{F}_q^l$ 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 more than one common intersection.

Now, let Γ denote the set of all straight lines that intersect with τ for at least $q^{3k/4}$ times. Thus, V' can be considered 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 intersection, m straight lines always will have less than m^2 intersections in total. Thus, if $|V'| \geq q^{3k/4}$, there would be more than $q^{3k/8}$ straight lines in Γ, i.e. we could find some $\Gamma' \subseteq \Gamma$, such that $|\Gamma'| = \lceil q^{3k/8}\rceil$. However, this will lead to a contradiction, as one can see as follws. Each of the straight lines in Γ' has less than $q^{3k/8}$ intersections with all other straight lines in Γ', what leaves more than $q^{3k/4} - q^{3k/8}$ intersections with τ that are not shared with other straight lines in Γ'. Hence, overall τ must have more than $\lceil q^{3k/8}\rceil \cdot (q^{3k/4} - q^{3k/8})$ of such non-shared intersections with straight lines in Γ', what directly contradicts the assumption that $q^k \leq q^{9k/8} - q^{3k/4}$. $\square$

Lemma 19. *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:*

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

Then it holds that $\Delta\big((\mathbf{z}_1, \ldots, \mathbf{z}_n), (\mathbf{u}_1, \ldots, \mathbf{u}_n)\big) < \frac{1}{2}\sqrt{\exp\big(nq^{2-k}\big) - 1}$.

Proof. We show this by very similar techniques as Lemma 15. 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 $(\mathbf{h}_1, \ldots, \mathbf{h}_n, \mathbf{z}_1, \ldots, \mathbf{z}_n)$. Now, some notes needed for our next estimation are in place. On the one hand, when $\mathbf{h}_i$ and $\mathbf{h}'_i$ are linearly independent, exactly q^{k-2} different $z \in \mathcal{U}$ will fulfill the following linear equation system:

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

On the other hand, when $\mathbf{h}_i$ and $\mathbf{h}'_i$ are linearly dependent, this linear equation system may have either q^{k-1} solutions or no solution at all—remember that always $\mathbf{h}_i, \mathbf{h}'_i \neq 0$, since $0 \notin \mathcal{H}$. Thus, using the auxiliary random variable $\mathbf{m} := \#\big\{ i \in \{1, \ldots, n\} \;\big|\; \mathbf{h}_i \text{ and } \mathbf{h}'_i \text{ are linearly independent} \big\}$ we can estimate:

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

In other words, it holds:

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

By the Pythagorean Theorem follows:

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

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:

$$\Delta\big((\mathbf{z}_1, \ldots, \mathbf{z}_n), (\mathbf{u}_1, \ldots, \mathbf{u}_n)\big) \;=\; \tfrac{1}{2}\|\vec{z} - \vec{u}\|_1 \;\leq\; \tfrac{1}{2}\sqrt{|\mathcal{U}^n|} \cdot \|\vec{z} - \vec{u}\|_2 \;\leq\; \tfrac{1}{2}\sqrt{\Big(1 + \tfrac{(q-1)^2}{|\mathcal{H}|}\Big)^n - 1}$$

To conclude our proof, we further estimate:

$$\Big(1 + \tfrac{(q-1)^2}{|\mathcal{H}|}\Big)^n \;=\; \exp\Big(n \cdot \ln\Big(1 + \tfrac{(q-1)^2}{|\mathcal{H}|}\Big)\Big) \;<\; \exp\Big(\tfrac{n \cdot (q-1)^2}{|\mathcal{H}|}\Big) \;<\; \exp\Big(nq^{2-k}\Big) \qquad \square$$

Corollary 20. *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, let $\mathbf{R}$ be some arbitrary random variable with finite support $\mathcal{R}$. For $i = 1, \ldots, n$ let any mapping $x_i : \mathcal{R} \times \mathcal{H}^n \times \mathcal{U}^{i-1} \to \mathbb{F}_q$ be given. Finally, for $i = 1, \ldots, n$ we define the following random variables:*

$$\mathbf{h}_i \overset{\mathrm{r}}{\leftarrow} \mathcal{H} \qquad\qquad \mathbf{z}_i \overset{\mathrm{r}}{\leftarrow} \big\{z \in \mathcal{U} \bigm| z\mathbf{h}_i = x_i(\mathbf{R}, \mathbf{h}_1, \ldots, \mathbf{h}_n, \mathbf{z}_1, \ldots, \mathbf{z}_{i-1})\big\}$$

Then it holds that $\iota\big(\mathbf{R}, (\mathbf{z}_1, \ldots, \mathbf{z}_n)\big) < \sqrt{\exp\big(nq^{2-k}\big) - 1}$.

Proof. Let $(\tilde{\mathbf{z}}_1, \ldots, \tilde{\mathbf{z}}_n)$ be identically distributed as $(\mathbf{z}_1, \ldots, \mathbf{z}_n)$, but be independent of $\mathbf{R}$, i.e. $\iota\big(\mathbf{R}, (\mathbf{z}_1, \ldots, \mathbf{z}_n)\big) = \Delta\big((\mathbf{R}, \mathbf{z}_1, \ldots, \mathbf{z}_n), (\mathbf{R}, \tilde{\mathbf{z}}_1, \ldots, \tilde{\mathbf{z}}_n)\big)$. Further, let $\mathbf{u}_1, \ldots, \mathbf{u}_n \overset{\mathrm{r}}{\leftarrow} \mathcal{U}$. First note that $\Delta\big((\mathbf{z}_1, \ldots, \mathbf{z}_n), (\mathbf{u}_1, \ldots, \mathbf{u}_n) \bigm| \mathbf{R} = R\big) < \frac{1}{2}\sqrt{\exp\big(nq^{2-k}\big) - 1}$ for every $R \in \mathcal{R}$ by Lemma 19. Moreover, it holds (see Section 2.1 for the definition of the conditioned statistical distance):

$$\Delta\big((\mathbf{R}, \mathbf{z}_1, \ldots, \mathbf{z}_n), (\mathbf{R}, \mathbf{u}_1, \ldots, \mathbf{u}_n)\big) \;=\; \sum_{R \in \mathcal{R}} \mathbb{P}[\mathbf{R} = R] \cdot \Delta\big((\mathbf{z}_1, \ldots, \mathbf{z}_n), (\mathbf{u}_1, \ldots, \mathbf{u}_n) \bigm| \mathbf{R} = R\big)$$

Hence we can estimate:

$$\Delta\big((\mathbf{R}, \mathbf{z}_1, \ldots, \mathbf{z}_n), (\mathbf{R}, \mathbf{u}_1, \ldots, \mathbf{u}_n)\big) \;<\; \tfrac{1}{2}\sqrt{\exp\big(nq^{2-k}\big) - 1}$$

Note that also $\Delta\big((\mathbf{R}, \tilde{\mathbf{z}}_1, \ldots, \tilde{\mathbf{z}}_n), (\mathbf{R}, \mathbf{u}_1, \ldots, \mathbf{u}_n)\big) < \frac{1}{2}\sqrt{\exp\big(nq^{2-k}\big) - 1}$, since by construction $\Delta\big((\mathbf{R}, \tilde{\mathbf{z}}_1, \ldots, \tilde{\mathbf{z}}_n), (\mathbf{R}, \mathbf{u}_1, \ldots, \mathbf{u}_n)\big) = \Delta\big((\mathbf{z}_1, \ldots, \mathbf{z}_n), (\mathbf{u}_1, \ldots, \mathbf{u}_n)\big)$ and by Lemma 19 we have that $\Delta\big((\mathbf{z}_1, \ldots, \mathbf{z}_n), (\mathbf{u}_1, \ldots, \mathbf{u}_n)\big) < \frac{1}{2}\sqrt{\exp\big(nq^{2-k}\big) - 1}$. Thus, using the Triangle Inequality we can finally conclude:

$$\Delta\big((\mathbf{R}, \mathbf{z}_1, \ldots, \mathbf{z}_n), (\mathbf{R}, \tilde{\mathbf{z}}_1, \ldots, \tilde{\mathbf{z}}_n)\big) \;<\; \sqrt{\exp\big(nq^{2-k}\big) - 1} \qquad \square$$

5 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.

5.1 Lower bounds for David's communication overhead

Even our refined construction for l-bit string-OT (q.v. Section 3.2.3) 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.

5.2 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 4.3). Instead, we have only a polynomial bound for the *expected* running time (cf. Lemma 4). 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, what was 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) by that one could avoid repeated token rewinding.

5.3 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 3.1 and Section 2.4). We will 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. we have that b and B are polynomially bounded in the security parameter. The corrupted token issuer randomly picks $j \xleftarrow{r} \{b, \dots, B\}$ 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.

5.4 Conclusion & improvement opportunities

In this paper, 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 by that one can 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. After all, we consider our work a substantial gain towards practical two-party computation, but still want to point out some issues that in our view 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 4). In contrast, we only need $\mathbb{F}_{2^l}^2$-OAFE for our optimal l-bit string-OT protocol (cf. Section 3.2.3) and are not aware of any 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 6) 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, we are not aware of any potential attack, 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, we believe that the blow-up factor can be reduced from 100 to 6 just by more sophisticated proof techniques and a slight modification of the protocol.

Less interaction. Our protocol $\Pi_{\text{OAFE}}^{\text{semi-int}}$ (q.v. Figure 6 in Section 3.1) 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 3.2.1). 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 we suggest that David just sends a random seed of a pseudorandom number generator. We believe (but were not able to prove) that this does not breach security, as long as 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 6 in Section 3.1) 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, we 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 3.2.6), 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 2.4). 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 3.2.3). 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.

References

[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.

[BBR88] Charles H. Bennett, Gilles Brassard, and Jean-Marc Robert. Privacy amplification by public discussion. *SIAM J. Comput.*, 17(2):210–229, 1988.

[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.

[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 version online available at `http://eprint.iacr.org/2000/067`.

[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.

[CKS$^+$11] Seung Geol Choi, Jonathan Katz, Dominique Schrder, Arkady Yerukhimovich, and Hong-Sheng Zhou. (Efficient) universally composable two-party computation using a minimal number of stateless tokens. Cryptology ePrint Archive, Report 2011/689, 2011.

[Cle91] Richard Cleve. Towards optimal simulations of formulas by bounded-width programs. *Computational Complexity*, 1:91–105, 1991.

[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.

[DKMQ12] Nico Döttling, Daniel Kraschewski, and Jörn Müller-Quade. Statistically secure linear-rate dimension extension for oblivious affine function evaluation. In *Information Theoretic Security, Proceedings of ICITS*, 2012. To appear.

[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.

[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.

[GO96] Oded Goldreich and Rafail Ostrovsky. Software protection and simulation on oblivious rams. *J. ACM*, 43(3):431–473, 1996.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom 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.

[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.

[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.

[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.

[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.