General Statistically Secure Computation with Bounded-Resettable Hardware Tokens

Nico Döttling*†[1], Daniel Kraschewski‡[2], Jörn Müller-Quade[3] and Tobias Nilges[3]

[1]Aarhus University, Denmark, nico.doettling@cs.au.dk
[2]TNG Technology Consulting GmbH, Munich, Germany, kraschew@ira.uka.de
[3]Karlsruhe Institute of Technology, Germany, {mueller-quade,nilges}@kit.edu

Abstract

Universally composable secure computation was assumed to require trusted setups, until it was realized that parties exchanging (untrusted) tamper-proof hardware tokens allow an alternative approach (Katz; EUROCRYPT 2007). This discovery initialized a line of research dealing with two different types of tokens. Using only a single *stateful* token, one can implement general statistically secure two-party computation (Döttling, Kraschewski, Müller-Quade; TCC 2011); though all security is lost if an adversarial token receiver manages to physically reset and rerun the token. *Stateless* tokens, which are secure by definition against any such resetting-attacks, however, do provably not suffice for statistically secure computation in general (Goyal, Ishai, Mahmoody, Sahai; CRYPTO 2010).

We investigate the natural question of what is possible if an adversary can reset a token at most a bounded number of times (e.g., because each resetting attempt imposes a significant risk to trigger a self-destruction mechanism of the token). Somewhat surprisingly, our results come close to the known positive results with respect to non-resettable stateful tokens. In particular, we construct polynomially many instances of statistically secure and universally composable oblivious transfer, using only a constant number of tokens. Our techniques have some abstract similarities to previous solutions, which we grasp by defining a new security property for protocols that use oracle access. Additionally, we apply our techniques to zero-knowledge proofs and obtain a protocol that achieves the same properties as bounded-query zero-knowledge PCPs (Kilian, Petrank, Tardos; STOC 1997), even if a malicious prover may issue *stateful* PCP oracles.

Keywords: resettable tamper-proof hardware, universal composability, statistical security, commitments, oblivious transfer, zero-knowledge

*Supported by European Research Commission Starting Grant no. 279447.

†The authors acknowledge support from the Danish National Research Foundation and The National Science Foundation of China (under the grant 61061130540) for the Sino-Danish Center for the Theory of Interactive Computation, within which part of this work was performed; and also from the CFEM research center (supported by the Danish Strategic Research Council) within which part of this work was performed.

‡Work done while at Technion, Israel. Supported by the European Union's Tenth Framework Programme (FP10/2010-2016) under grant agreement no. 259426 – ERC Cryptography and Complexity.

Contents

1 Introduction

The model of untrusted tamper-proof hardware was introduced by [Kat07] to circumvent trusted setup assumptions and has proven to be a strong tool for creating cryptographic protocols, especially in the context of universally composable (UC-secure) [Can01] multi-party computation. In the tamper-proof hardware model, (possibly malicious) parties can create tokens and send them to other parties, who then can interact with the tokens but not access any internal secrets.

In this line of research, there are two different types of tokens considered: *stateful* and *stateless/resettable* tokens. Studies of the latter are usually motivated by so-called *resetting-attacks*, meaning that an adversarial receiver could physically reset a token's internal state (e.g., by cutting off the power supply). [DKMQ11] implemented multiple instances of statistically UC-secure oblivious transfer (OT), using only a single stateful token. On the downside, [GIMS10] showed that with any number of stateless tokens statistical OT is impossible, even if one only goes for stand-alone security. In fact, only a few statistically secure protocols based on stateless tokens have been proposed, namely stand-alone secure commitments [GIMS10] and a UC-secure variant [DS13], the latter using bidirectional exchange of polynomially many tokens for each commitment. These positive results are complemented again by [GIMS10], showing that unconditional non-interactive commitments cannot be performed by using only stateless tokens. Since all known approaches based on stateful tokens completely break down if only a single resetting attempt is successful, and strong impossibility results hold with respect to arbitrarily resettable tokens, it seems a natural question what is still possible if an a priori bound for successful resettings is known. Therefore, similar in nature to the well-studied problems of bounded leakage [KV09, DKW11], bounded-resettable zero-knowledge [KP01, MR01, ZDLZ03, BLV06], and bounded-query zero-knowledge PCPs [KPT97, IMS12], we propose a bounded-resettable hardware model.

Our new model can also be seen as a variant of the PCP model [FGL$^+$91, AS92] or interactive PCP model [KR08], depending on whether a considered protocol contains direct interaction between the token issuer and the token receiver. The difference to the (interactive) PCP model is that maliciously issued tokens/oracles can be stateful. This seems reasonable, since it is hard to verify that a malicious token is stateless. We show that this weakened version of the PCP model still allows non-interactive zero-knowledge with $O(1)$ rounds of oracle queries and even general (interactive) secure computation. [GIMS10] also considered stateful malicious PCP oracles, though without an a priori query bound. They constructed in that model interactive zero-knowledge proofs with non-constant round complexity and showed impossibility of general statistically secure computation. Moreover, the result on ZK-PCPs by [KPT97] can be made robust against malicious stateful oracles straightforwardly at the cost of issuing polynomially many oracles and have the verifier query each oracle only once. It is not clear, however, if the same result can be obtained by using only a constant amount of oracles.

Our Results. We define a *bounded-resettable hardware model* and achieve in this model to a large extent the known positive results for stateful tokens. This is surprising, because the corresponding stateful-token protocols from the literature are all susceptible to resetting-attacks. We construct

- multiple commitments, based on a single token issued by the commitment sender,
- a single string-commitment, based on a token issued by the commitment receiver,
- multiple instances of OT, based on $O(1)$ tokens issued by the OT-sender, and
- a bounded-resettable zero-knowledge proof of knowledge, with $O(1)$ tokens from the prover.

All protocols are statistically UC-secure and efficient. The first commitment protocol can be made non-interactive, sacrificing UC-security against a corrupted sender, remaining statistically binding. The zero-knowledge protocol can be implemented such that the verifier does not communicate with

the prover but only with tokens sent by the prover. Moreover, if we assume that even malicious provers can only issue stateless tokens, then all token functionalities in the zero-knowledge protocol can be combined on a single token and we end up with the same result as [KPT97].

Our Techniques. The main technical difficulty we have to deal with is that a malicious token issuer can store an arbitrarily complicated function an a token. We enforce (to some extent) honest programming by a simple challenge-response protocol. The domain of allowed token functionalities is chosen such that it is a linear space. The token receiver announces a random linear projection and the token issuer has to reveal the token functionality under this projection. The receiver can then check if the token reacts consistently, while learning only part of the function parameters. We put forward an abstract notion of this technique, which we call *oracle validation*. It has previously been used in a more ad-hoc manner by [DKMQ11, CKS+14], though their space of token functions is quite different from ours: They use affine functions that map length-n bit-vectors to $(n \times n)$-matrices, whereas we use higher-degree univariate polynomials that operate on a large finite field.

The composability proof for one of our commitment schemes also requires some constructive algebraic geometry, namely efficient uniform sampling from large finite varieties [CS09].

Related Work. The notion of resettable zero-knowledge was introduced by [CGGM00]. In this model, a malicious verifier is allowed to reset the prover arbitrarily and rerun the protocol. Constant-round black-box zero-knowledge protocols with resettable provers were only achieved in various public key models where the verifier has to register a public key, like Bare Public Key Model [CGGM00, CPV04a, YZ07, DL08], Upperbounded Public Key (UPK) Model [ZDLZ03], Weak Public Key (WPK) Model [MR01] and Counter Public Key Model [CPV04b]. UPK and WPK assume that the amount of resets is a priori bounded, similar to our model. [BGGL01] provided the first construction of a (non-black-box) resettably-sound zero-knowledge argument system, where soundness for a resettable verifier is achieved. This work was later improved [BP13, CPS13] and generalized to simultaneously resettable zero-knowledge protocols [DGS09, DFG+11, COSV12, BP13, COPV13]. Since then resettability has found its way into general multi-party computation [GS09, GM11].

Early works concerning tamper-proof hardware made computational assumptions and assumed stateful tokens [Kat07, GKR08]. This was later relaxed to resettable or stateless tokens [CGS08, DMMQN13, CKS+14] and/or unconditional security [MS08, GIS+10, GIMS10, DKMQ11, DS13].

2 Preliminaries

2.1 The UC-framework

We state and prove the security of our protocols in the Universal Composability (UC) framework of [Can01]. In this framework security is defined by comparison of a *real model* and an *ideal model*. The protocol of interest Π is running in the former, where an adversary $\mathcal{A}$ coordinates the behavior of all corrupted parties. We assume static corruption, i.e., the adversary $\mathcal{A}$ cannot adaptively change corruption during a protocol run. 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. Denote the random variable representing the output of $\mathcal{Z}$ when interacting with the real model by $\mathsf{Real}^{\Pi}_{\mathcal{A}}(\mathcal{Z})$ and when interacting with the ideal model by $\mathsf{Ideal}^{\mathcal{F}}_{\mathcal{S}}(\mathcal{Z})$. Protocol Π is said to UC-implement $\mathcal{F}$, if for every adversary $\mathcal{A}$ there exists a simulator $\mathcal{S}$, such that for all environments $\mathcal{Z}$ the distributions of $\mathsf{Real}^{\Pi}_{\mathcal{A}}(\mathcal{Z})$ and $\mathsf{Ideal}^{\mathcal{F}}_{\mathcal{S}}(\mathcal{Z})$ are indistinguishable. Since we aim at statistical security, all entities are computationally unbounded. However, the (expected) runtime complexity of the ideal model has to be polynomial in the runtime complexity of the real model.

Protocol $\Pi^{\mathrm{val}}_{k\text{-ind}}$

Implicitly parametrized by a finite field $\mathbb{F}$ and a dimension $n \in \mathbb{N}$. The sender's input domain consists of all polynomials $p \in \mathbb{F}^n[X]$ of degree at most $k-1$. The security parameter is $\ell := \log|\mathbb{F}|$.

1. Sender: Let $p \in \mathbb{F}^n[X]$ be the sender's input. Pick $p' \in \mathbb{F}^n[X]$ of degree at most $k-1$ uniformly at random. Program the oracle such that on input $x \in \mathbb{F}$ it outputs $\big(p(x), p'(x)\big)$.
2. (a) Receiver: Pick $\lambda \in \mathbb{F}$ uniformly at random and send it to the sender.
   (b) Sender: Compute $\tilde{p} := \lambda \cdot p + p'$ and send it to the receiver.
3. Receiver: Let $x \in \mathbb{F}$ be the receiver's input. Input x into the oracle; let (y, y') denote the response.
4. Receiver: Verify that $\deg(\tilde{p}) \leq k-1$ and $\lambda \cdot y + y' = \tilde{p}(x)$. If so, output y; otherwise reject.

Figure 1: Construction of a query-once validation scheme for a k-wise independent oracle.

2.2 Definitions and notations

We write $\Delta(x, y)$ for the statistical distance between x and y. The inner product of x, y is denoted as $\langle x \,|\, y \rangle$ and their concatenation as $x \| y$. By $\mathbb{F}_q$ we denote the finite field with q elements.

We canonically extend the notion of polynomials over a field $\mathbb{F}$ as follows. By $\mathbb{F}^n[X]$ we denote the set of all n-tuples of polynomials $p_1, \ldots, p_n \in \mathbb{F}[X]$. Each polynomial $p := (p_1, \ldots, p_n) \in \mathbb{F}^n[X]$ defines a function $\mathbb{F} \to \mathbb{F}^n$, $x \mapsto \big(p_1(x), \ldots, p_n(x)\big)$, whose degree is $\deg(p) := \max_{i=1}^{n}(\deg(p_i))$. We treat $\mathbb{F}^n[X]$ as an $\mathbb{F}$-linear vector space in the natural way.

Further notations are only used in the appendices and are introduced there. All (close to) standard ideal functionalities for the UC-framework can be found in Appendix A.

3 Query-Once Oracle Validation

We introduce now our abstract notion of enforcing honest token programming. Consider a scenario consisting of an honest receiver party, a (possibly) malicious sender party, and an oracle which is arbitrarily programmable by the sender in a setup phase. All entities are computationally unbounded. The security feature we aim at, is that the sender has to choose the oracle functionality from some predefined class and otherwise is caught cheating, even though the receiver queries the oracle only once. If this is achieved, we speak of a *query-once oracle validation scheme*. More particularly, such a scheme consists of four stages (for a concrete example protocol, where the domain of allowed functions consists of bounded degree polynomials over a finite field, see Figure 1):

1. The sender programs the oracle.
2. Sender and receiver run an interactive protocol which is independent of the receiver's input.
3. The receiver chooses his input and queries the oracle.
4. The receiver either rejects or produces some output.

Let g denote the sender input and x the receiver input. We require the following properties.

Efficiency: All computations by honest entities have polynomial complexity.

Correctness: If the sender is honest, then the receiver always outputs $g(x)$ and never rejects.

Privacy: The receiver does not learn anything else about g than $g(x)$.

Extractability: Even if the sender is corrupted, an extractor Ext with access to the oracle program T^* and the message transcript τ of Stage 2 can compute a valid sender input g such that a receiver R with uniformly random input x with overwhelming probability (taken over the

randomness of x and all of R's and Ext's random choices) either rejects or outputs $g(x)$. The extractor has to be efficient in the sense that its expected runtime on any input (τ, T^*) is asymptotically bounded by $(\ell \cdot |\mathsf{T}^*|)^{O(1)} \cdot \rho^{-1}$, where ℓ is a security parameter, $|\mathsf{T}^*|$ is the size of the oracle program, and ρ is R's accept probability conditioned on τ (still with random x).

Note that $g(x')$ can be information-theoretically reconstructed from the receiver's view for any input x' that matches his oracle query. It follows by the privacy property that his input x must be uniquely determined by his message to the oracle. Thus, w.l.o.g. he just sends x to the oracle.

Next, we show that the oracle validation scheme $\Pi^{\text{val}}_{k\text{-ind}}$ is indeed extractable—efficiency, correctness, and privacy are straightforward to see. The extractor construction is the main ingredient for our upcoming UC proofs.

Lemma 3.1. *Figure 1 describes an oracle validation scheme. In particular, there exists an extractor* Ext*, such for every pair* $(\mathsf{S}^*, \mathsf{T}^*)$ *of a corrupted sender* S^* *and a corrupted oracle* T^* *it holds:*

- *Provided arbitrarily rewindable access to* T^* *and given a transcript* $\tau = (\lambda, \tilde{p})$ *of the messages between* S^* *and an honest receiver* R *(i.e., with uniformly random* $\lambda \in \mathbb{F}$*),* Ext *computes a polynomial* $p \in \mathbb{F}^n[X]$ *of degree at most* $k-1$*.*
- *If* R*'s input* x *is uniformly random, then with some overwhelming probability* $1-\rho'$ *(taken over the randomness of* λ*,* x*, and* Ext*'s random tape),* R *either rejects or outputs* $p(x)$*. In particular, we have a failure probability* $\rho' \leq |\mathbb{F}|^{-\Omega(1)}$*.*
- *For every possible transcript* τ*, the expected number of queries from* Ext *to* T^* *is* $k \cdot \rho^{-1}$*, where* ρ *is* R*'s accept probability conditioned on* τ *and averaged over all inputs* $x \in \mathbb{F}$*. The rest of* Ext*'s calculations have an overall time complexity which is polynomial in* $n, k, \log|\mathbb{F}|$*.*

Proof. The extractor Ext runs a simple trial-and-error approach. It repeatedly samples a uniformly random oracle input $x \in \mathbb{F}$, until it has found inputs $x_1, \ldots, x_k$ such that the corresponding oracle outputs $(y_i, y_i') := \mathsf{T}^*(x_i)$ pass the consistency checks $\lambda \cdot y_i + y_i' \stackrel{?}{=} \tilde{p}(x_i)$. Then, Ext computes and outputs the minimal-degree interpolation polynomial $p \in \mathbb{F}^n[X]$ with $p(x_i) = y_i$ for $i = 1, \ldots, k$. Note that we do not enforce pairwise distinctness of $x_1, \ldots, x_k$.

There are two things to show. Firstly, we have to show that R on random input x basically either rejects or produces output $p(x)$. Secondly, we have to estimate the expected number of queries from Ext to T^*. We start with the latter. The sampling of each x_i is a stochastic process with geometric distribution of the number of oracle queries: Given that ρ is R's accept probability conditioned on some transcript τ, the expected number of queries for sampling one x_i is $\sum_{j=1}^{\infty} j \cdot (1-\rho)^{j-1} \cdot \rho = \rho^{-1}$. The sampling of $x_1, \ldots, x_k$ hence requires $k \cdot \rho^{-1}$ queries to T^* on average.

Next, we turn to the question of how well the extracted polynomial p approximates the functionality of a real protocol run. W.l.o.g., S^* follows a deterministic worst-case strategy and we can consider it as a function that maps each possible challenge $\lambda \in \mathbb{F}$ to a polynomial $\tilde{p}_\lambda \in \mathbb{F}^n[X]$ with $\deg(\tilde{p}_\lambda) \leq k-1$. Analogously, T^* implements a deterministic function by assumption. Thus, for each combination of λ and x it is fixed whether R finally rejects or not. This defines a relation between challenges λ and oracle inputs x. It can be represented as a bipartite graph, where a left-hand vertex λ is adjacent to a right-hand vertex x if R does not reject the corresponding protocol run. Our proof now boils down to showing that there exists a subset of "bad" edges E' such that

1. uniformly random λ and x are adjacent via a "bad" edge only with negligible probability, namely $|E'|/|\mathbb{F}|^2 \leq |\mathbb{F}|^{-\Omega(1)}$, and
2. after removal of all "bad" edges from the graph, T^* implements on each neighborhood of a possible challenge λ a polynomial function of degree at most $k-1$.

Functionality $\mathcal{F}_{\text{wrap}}^{\text{b-r}}$

Implicitly parametrized by a query bound q. The variable *resets_left* is initialized by *resets_left* $\leftarrow q-1$.

Creation:

1. Await an input $(\texttt{create}, \mathcal{M}, b)$ from the token issuer, where $\mathcal{M}$ is a deterministic Turing program and $b \in \mathbb{N}$. Then, store $(\mathcal{M}, b)$ and send $(\texttt{created})$ to the adversary.
2. Await a message $(\texttt{delivery})$ from the adversary. Then send $(\texttt{ready})$ to the token receiver.

Execution:

3. Await an input $(\texttt{run}, w)$ from the receiver. Run $\mathcal{M}$ on input w. When $\mathcal{M}$ halts without generating output or b steps have passed, send a special symbol $\perp$ to the receiver; else send the output of $\mathcal{M}$.

Reset (adversarial receiver only):

4. Upon receiving a message $(\texttt{reset})$ from a corrupted token receiver, verify that *resets_left* > 0. If so, decrease *resets_left* by 1 and go back to Step 3; otherwise ignore that message.

Figure 2: The wrapper functionality by which we model bounded-resettable tamper-proof hardware. The runtime bound b is merely needed to prevent malicious token senders from providing a perpetually running program code $\mathcal{M}$; it will be omitted throughout the rest of the paper.

For the existence proof of E' see Appendix B (Corollary B.5). The key observations used there are

- that T^* implements a polynomial function of low degree on the common neighborhood $\mathcal{N}(\lambda) \cap \mathcal{N}(\lambda')$ of any distinct challenges λ, λ' and
- that after removal of only a few edges, our graph decomposes into a disjoint collection of complete bipartite subgraphs.

Once E' is shown to exist, we finally need to argue that the following event has probability $|\mathbb{F}|^{-\Omega(1)}$:

- The receiver does not reject and
- one of the oracle inputs $x_1, \ldots, x_k$ sampled by Ext is adjacent via a "bad" edge to the challenge λ given by τ, or $x_i = x_j$ for some $i \neq j$.

It then follows that $\rho' \leq |\mathbb{F}|^{-\Omega(1)}$. However, since $|E'|/|\mathbb{F}|^2 \leq |\mathbb{F}|^{-\Omega(1)}$, we already have with probability $1-|\mathbb{F}|^{-\Omega(1)}$ (taken over the randomness of λ) that the given challenge λ is only adjacent to an $|\mathbb{F}|^{-\Omega(1)}$-fraction of all inputs $x \in \mathbb{F}$ or λ is adjacent to $|\mathbb{F}|^{\Omega(1)}$ edges of which only an $|\mathbb{F}|^{-\Omega(1)}$-fraction is "bad". This implies that the event above has the claimed negligible probability. □

4 Bounded-Resettable Tamper-Proof Hardware

In this section we define and discuss the ideal functionality for bounded-resettable tamper-proof hardware (q.v. Figure 2). It is a slightly modified version of the $\mathcal{F}_{\text{wrap}}$-functionality introduced by [Kat07]. The token sender provides a (w.l.o.g. deterministic) Turing machine and the receiver can then run it once on an input word of his choice, staying oblivious of any internal secrets. A malicious receiver can reset the token and query it repeatedly, until some bound q is reached and the functionality does not respond any more. The query bound q models an estimation for how often an adversary could reset a token that is meant to shut down for good after the first query. All our protocols rely on q being polynomially bounded in the security parameter. A smaller bound q implies better efficiency.

We stress that tokens are not actually required to contain a state that counts the number of queries. Our definition of $\mathcal{F}_{\text{wrap}}^{\text{b-r}}$ is just the most general way to model *any* kind of token for which an upper bound of resets can be derived. E.g., it suffices that $(1-\rho)^q$ is negligible, where ρ is an upper bound for the probability that the token successfully self-destructs after a query. As well, the token could try to delete its program $\mathcal{M}$ or make it inaccessible but an adversarial receiver could slow down that process or interrupt the deletion before it is complete, so that several queries are possible before $\mathcal{M}$ becomes finally out of reach for him. One can also imagine that security is only needed for some limited time (which is usually the case for the binding property of commitments) and hence it suffices to estimate the number of queries within this time. The latter seems particularly feasible, because it relies on the minimum possible response time of an honestly generated token.

Further note that our definition can be canonically extended to tokens that can be queried more than once also by honest users. However, our approach has the advantage to be trivially secure against tokens that maliciously change their functionality depending on the input history.

Our model is weaker than the stateful-token model in the sense that no previously known protocol with stateful tokens can tolerate even a single reset. They would be all completely broken. Therefore, none of the known positive results for stateful tokens does carry over to our model (unless $q = 1$). In turn, bounded-resettable tokens can be trivially implemented from unresettable stateful tokens. So, our results are strictly stronger than the corresponding results for stateful tokens. On the other hand, bounded-resettable tokens are strictly more powerful than arbitrarily resettable (i.e., standard stateless) tokens, since non-interactive commitments and statistically secure OT are possible with the former but impossible with the latter.

4.1 Commitments from the token sender to the token receiver

The basic idea how the token issuer can commit himself to some secret s is quite simple. He stores a random degree-q polynomial p on the token and sends the token together with $r := s + p(0)$ to the receiver. The token lets the receiver evaluate p on arbitrary challenges x, except for $x = 0$. To unveil s, the sender sends a description of p. The scheme is perfectly hiding, because even a corrupted receiver can query the token on at most q inputs, receiving only randomness that is statistically independent of $p(0)$. The scheme is statistically binding, because for any two distinct unveil messages p, p' and a uniformly random token input x it holds with overwhelming probability (namely at least $1 - \frac{q}{|\mathbb{F}|-1}$, where $\mathbb{F}$ is the finite field in which all computations take place) that $p(x) \neq p'(x)$ and thus at least one unveil message will be inconsistent with the receiver's view.

Unfortunately, the scheme as stated above is not UC-secure against a corrupted sender. The reason for this is that the sender simulator must be able to extract the secret s from the token program and the commit message r. If the token is issued honestly and thus implements a degree-q polynomial p, the simulator can evaluate the token code on $q+1$ different inputs, then reconstruct p, and compute $s = r - p(0)$. However, a maliciously issued token can implement an arbitrarily complicated function, which behaves like a degree-q polynomial only on a vanishing but still non-negligible fraction of inputs. It is at the very least unclear if one can extract the correct polynomial from such a token efficiently. Therefore, we employ our oracle validation scheme from Section 3 to make the token extractable. See Figure 3 for the resulting commitment protocol, which even implements many commitments using only one token.

Lemma 4.1. *The protocol $\Pi_{\text{COM}}^{\text{s-o}}$ (q.v. Figure 3) UC-implements $\mathcal{F}_{\text{COM}}^{\text{s-o}}$ (q.v. Appendix A.2).*

Proof-sketch. We start with the case of a corrupted receiver. The main issue in this case is that the simulator has to equivocate commitments in the unveil phase. He can do so by picking polynomials $\hat{p}, \hat{p}' \in \mathbb{F}_{2^\ell}^n[X]$ such that

Protocol $\Pi^{\text{s-o}}_{\text{COM}}$

Implicitly parametrized by a token query bound q, a commitment number n, and a commitment length ℓ. The security parameter is ℓ. For any vector $v = (v_1, \ldots, v_n)$ and $I \subseteq \{1, \ldots, n\}$ let $v_I := (v_i)_{i \in I}$.

Setup phase:

1. Sender: Pick two uniformly random polynomials $p, p' \in \mathbb{F}_{2^\ell}^n[X]$ of degree at most q. Program a token T which on input $x \in \mathbb{F}_{2^\ell} \backslash \{0\}$ outputs $\big(p(x), p'(x)\big)$ and ignores input $x = 0$. Send T to the receiver.
2. Receiver: Pick $\lambda \in \mathbb{F}_{2^\ell}$ uniformly at random and send it to the sender.
3. Sender: Compute $\tilde{p} := \lambda \cdot p + p'$ and send it to the receiver.

Commit phase:

4. Sender: Let $s := (s_1, \ldots, s_n) \in \mathbb{F}_{2^\ell}^n$ be the sender's input. Send $r := s + p(0)$ to the receiver.
5. Receiver: Input a uniformly random $x \in \mathbb{F}_{2^\ell} \setminus \{0\}$ into T; let (y, y') denote the response.

Unveil phase:

6. Sender: Let $I \subseteq \{1, \ldots, n\}$ indicate the commitments to be opened. Send (I, p_I) to the receiver.
7. Receiver: If $\deg(\tilde{p}) \leq q$ and $\lambda \cdot y + y' = \tilde{p}(x)$ and $p_I(x) = y_I$, output $\hat{s}_I := r_I - p_I(0)$; else reject.

Figure 3: Statistically UC-secure commitment scheme where the sender is the token issuer.

- $\big(\hat{p}(x), \hat{p}'(x)\big) = \big(p(x), p'(x)\big)$ for all inputs x on which the simulated token was queried so far,
- $\lambda \cdot \hat{p} + \hat{p}' = \tilde{p}$, $\deg(\hat{p}, \hat{p}') \leq q$, and
- $\hat{p}_I(0) = \hat{s}_I$, where $\hat{s}_I$ is the desired result of the equivocation,

and reprogramming the token such that on input x it now outputs $\big(\hat{p}(x), \hat{p}'(x)\big)$. The unveil message for equivocating the commitment to $\hat{s}_I$ is just $(I, \hat{p}_I)$. Since the corrupted receiver can query the token at most q times, this is in his view perfectly indistinguishable from a proper commitment.

Now we show security against a corrupted sender. The simulator has to extract commitments in the unveil phase. He can do so by running the extractor Ext from Lemma 3.1 on the transcript of the setup phase and with rewindable access to the token code T^*. The extracted polynomial p allows the simulator to reconstruct the committed secret s from the corrupted sender's commit message r as $s = r - p(0)$. Note that Ext may have exponential runtime, but only needs to be run if by the end of the commit phase it is not already clear that the receiver will reject anyway. Therefore, the simulator must first check that $\lambda \cdot y + y' = \tilde{p}(x)$ and then run Ext only if the check is passed. Since Ext has complexity $\big(\ell \cdot |\mathsf{T}^*|\big)^{O(1)} \cdot \rho^{-1}$, where ρ is just the probability that this check is passed, we end up with an expected simulation complexity of $\big(\ell \cdot |\mathsf{T}^*|\big)^{O(1)}$. □

Remark 4.2. The commitment scheme $\Pi^{\text{s-o}}_{\text{COM}}$ is statistically binding, even if λ is fixed and known to the sender. This yields a statistically secure non-interactive commitment scheme in the bounded-resettable hardware model, which was proven impossible in the stateless-token model [GIMS10].

4.2 Commitments from the token receiver to the token sender

For a commitment from the token receiver to the token sender we need a slightly more sophisticated approach. As in our previous commitment scheme, the token implements a random degree-q polynomial p. The token receiver can then commit to some secret s by inputting a random x into

Protocol $\Pi^{\text{rev}}_{\text{COM}}$

Implicitly parametrized by a token query bound q and a commitment length ℓ. The security parameter is ℓ. Let $\sigma : \mathbb{F}_{2^{4\ell}} \to \mathbb{F}_{2^\ell}^4$, $x \mapsto \big(\sigma_1(x), \ldots, \sigma_4(x)\big)$ be the canonical $\mathbb{F}_{2^\ell}$-vector space isomorphism.

Setup phase:

1. Receiver: Pick two uniformly random polynomials $p, p' \in \mathbb{F}_{2^{4\ell}}[X]$ of degree at most q and program a token T which on input $x \in \mathbb{F}_{2^{4\ell}}$ outputs $\big(p(x), p'(x)\big)$. Send T to the sender.
2. Sender: Pick $\lambda \in \mathbb{F}_{2^{4\ell}}$ uniformly at random and send it to the receiver.
3. Receiver: Compute $\tilde{p} := \lambda \cdot p + p'$ and send it to the sender.

Commit phase:

4. Sender: Let $s \in \mathbb{F}_{2^\ell}$ be the sender's input. Input a uniformly random $x \in \mathbb{F}_{2^{4\ell}}$ into the token T; let (y, y') denote the response. If $\lambda \cdot y + y' = \tilde{p}(x)$ and $\deg(\tilde{p}) \leq q$, pick a uniformly random $h \in \mathbb{F}_{2^\ell}^4$ and compute $m := s + \langle h \,|\, \sigma(x) \rangle$ and $\tilde{y} := \sigma_1(y)$ and send $(m, h, \tilde{y})$ to the receiver; otherwise abort.

Unveil phase:

5. Sender: Send x to the receiver.
6. Receiver: Verify that $\tilde{y} = \sigma_1(p(x))$. If so, output $\hat{s} := m - \langle h \,|\, \sigma(x) \rangle$; otherwise reject.

Figure 4: Statistically UC-secure commitment scheme where the receiver is the token issuer.

the token, thus learning $p(x)$, and announcing a commit message that consists of

- a fraction of bits of $p(x)$, say the first quarter of its bit-string representation,
- a 2-universal hash function h, and
- $m := s + h(x)$.

To unveil s, he just needs to announce the used token input x. We briefly sketch now why this scheme is hiding and binding. We start with the latter. Due to the query-bound q, the token acts just like a perfectly random function. Thus, a corrupted commitment sender may only with negligible probability find two distinct unveil messages x, x' such that $p(x)$ and $p(x')$ agree on the first quarter of their bit-string representation. This establishes the binding property. The token issuer, however, learns only several bits of information about x during the commit phase, so that from his view x has still linear entropy afterwards. Since h is a 2-universal hash function, this means that he cannot predict $h(x)$ and thus the commitment is hiding. Still, we need to employ our oracle validation scheme from Section 3 again to make the token extractable, as otherwise we have no UC-security against a corrupted commitment receiver. See Figure 4 for the resulting protocol.

Lemma 4.3. *The protocol $\Pi^{\text{rev}}_{\text{COM}}$ (q.v. Figure 4) implements $\mathcal{F}_{\text{COM}}$ (q.v. Appendix A.1) UC-secure against a corrupted commitment sender.*

Proof-sketch. We just have to exploit that the simulator sees all token inputs. As the number of token queries by the commitment sender is upper bounded by q, the token acts from his views like a perfectly random function. Hence, with overwhelming probability his announcement of $\tilde{y}$ in the commit phase either corresponds to a unique input x already sent to the token or he is caught cheating in the unveil phase. In the former case, the simulator can find x just by scanning through the token's input history, compute the correct secret $s = m - \langle h \,|\, \sigma(x) \rangle$ and send it to the ideal commitment functionality $\mathcal{F}_{\text{COM}}$. In the other case, the simulator can just send anything to the ideal functionality, because only with negligible probability he might need to unveil it later. □

Simulator for a corrupted token issuer that receives commitments

Setup phase: Simulated straightforwardly, using a simulated version of the complete real model where the simulated adversary is wired to the ideal model's environment in the canonical way. Store $(\lambda, \tilde{p})$ and the token program T^* sent by the corrupted commitment receiver to the simulated functionality $\mathcal{F}_{\text{wrap}}^{\text{b-r}}$.

Commit phase: Simulated straightforwardly, with random sender input s. Store $(m, h, \tilde{y})$.

Unveil phase: If the simulated commitment sender has already aborted, do nothing. Otherwise, upon receiving $(\texttt{opened}, \hat{s})$ from $\mathcal{F}_{\text{COM}}$ replace the stored unveil information x in the simulated sender's memory with $\hat{x}$, computed by the following equivocation program, and let him then proceed with the protocol.

1. Setup the extractor Ext from Lemma 3.1 with parameters $\mathbb{F} := \mathbb{F}_{2^{4\ell}}$, $n := 1$, and $k := q+1$. Provide Ext with the transcript $\tau := (\lambda, \tilde{p})$ and rewindable access to the token code T^*.
2. Start Ext. If Ext queries T^* more than 2^ℓ times, give up; otherwise let p denote Ext's output.
3. Compute the unique polynomial $p_1 \in \mathbb{F}_{2^\ell}[X_1, \ldots, X_4]$ such that $\deg(p_1) \leq \deg(p)$ and $\sigma_1 \circ p = p_1 \circ \sigma$, where "$\circ$" denotes the function composition operator. Then pick a uniformly random solution $\hat{x} \in \mathbb{F}_{2^{4\ell}}$ of the following polynomial equation system, using the efficient algorithm of [CS09]:

$$\begin{aligned} p_1(\sigma(\hat{x})) &= \tilde{y} \\ \langle h \,|\, \sigma(\hat{x}) \rangle &= m - \hat{s} \end{aligned}$$

   Resample $\hat{x}$ until $p(\hat{x}) = t(\hat{x})$ and $\tilde{p}(\hat{x}) = \lambda \cdot t(\hat{x}) + t'(\hat{x})$, where $t, t' : \mathbb{F}_{2^{4\ell}} \to \mathbb{F}_{2^{4\ell}}$ such that $\mathsf{T}^*(\hat{x}) = \big(t(\hat{x}), t'(\hat{x})\big)$. Give up, if more than $2^{\sqrt{\ell}}$ iterations are required.
4. Replace x in the simulated sender's memory by $\hat{x}$.

Figure 5: Simulator for a corrupted token issuer in the protocol $\Pi_{\text{COM}}^{\text{rev}}$ (q.v. Figure 4).

Proving UC-security against a corrupted commitment receiver, i.e. providing a simulator that equivocates commitments, is more challenging. Note that even after extracting a polynomial p that approximates the token functionality, it is still nontrivial to find a token input $\hat{x}$ such that the first quarter of bits of $p(\hat{x})$ matches the given commit message $(m, h, \tilde{y})$ while $m - h(x) = \hat{s}$ for a new secret $\hat{s}$. This problem can be expressed as a polynomial equation system. Here the efficient algorithm of [CS09] for sampling random solutions comes into play. (See Appendix C for a brief explanation that all preconditions of [CS09, Theorem 1.1] are met.) In addition, the simulator has to make sure that the sampled solution $\hat{x}$ is actually possible in the real model: He has to (re)sample $\hat{x}$ until $p(\hat{x})$ agrees with the token functionality and the consistency check in Step 4 of the commit phase of $\Pi_{\text{COM}}^{\text{rev}}$ is passed. See Figure 5 for the detailed simulator description. The resampling of $\hat{x}$ imposes some extra difficulty for the runtime estimation, but we postpone the technical calculation to Appendix D. Next, we show that our scheme is statistically hiding. This is needed for the UC proof and has further application later in our construction of resettable zero-knowledge.

Lemma 4.4. *The protocol $\Pi_{\text{COM}}^{\text{rev}}$ is statistically hiding, even if λ is fixed.*

Proof. Let λ and $\tilde{p}$ be arbitrary but fixed. Let $t, t' : \mathbb{F}_{2^{4\ell}} \to \mathbb{F}_{2^{4\ell}}$ represent the (possibly) corrupted token functionality in the sense that the token maps $x \mapsto \big(t(x), t'(x)\big)$. Moreover, let $Z := \mathbb{F}_{2^\ell} \cup \{\perp\}$ and for each $z \in Z$ let M_z denote the set of all token inputs x that lead to a commit message $(m, h, \tilde{y})$ with $\tilde{y} = z$. I.e., $M_z = \{x \in \mathbb{F}_{2^{4\ell}} \,|\, \lambda \cdot t(x) + t'(x) = \tilde{p}(x) \,\wedge\, \sigma_1(t(x)) = z\}$ for $z \in \mathbb{F}_{2^\ell}$ and $M_\perp = \{x \in \mathbb{F}_{2^{4\ell}} \,|\, \lambda \cdot t(x) + t'(x) \neq \tilde{p}(x)\}$. For uniformly random $x \in \mathbb{F}_{2^{4\ell}}$ and the corresponding $\tilde{y}$

(meaning that $\tilde{y} = \sigma_1(t(x))$ if $\lambda \cdot t(x) + t'(x) = \tilde{p}(x)$ and else $\tilde{y} = \perp$) it holds:

$$\max_{e:Z\to\mathbb{F}_{2^{4\ell}}} \Pr[x = e(\tilde{y})] \;=\; \mathrm{E}(|M_{\tilde{y}}|^{-1}) \;=\; \sum\nolimits_{z\in Z} \Pr[x \in M_z]\cdot|M_z|^{-1} \;=\; \sum\nolimits_{z\in Z} \tfrac{1}{|\mathbb{F}_{2^{4\ell}}|} \;=\; 2^{-3\ell}+2^{-4\ell}$$

Hence, for uniformly random $u \in \mathbb{F}_{2^\ell}$ we can conclude by the Leftover hash lemma (Lemma E.1):

$$\Delta\big((\langle h \,|\, \sigma(x)\rangle, h, \tilde{y}), (u, h, \tilde{y})\big) \;\le\; \tfrac{1}{2}\sqrt{\max_{e:Z\to\mathbb{F}_{2^{4\ell}}} \Pr[x = e(\tilde{y})] \cdot |\mathbb{F}_{2^\ell}|} \;=\; \tfrac{1}{2}\sqrt{2^{-2\ell} + 2^{-3\ell}} \;<\; 2^{-\ell}$$

It directly follows now that the statistical distance between a commitment on any secret s and a commitment on uniform randomness is also upper bounded by $2^{-\ell}$. □

Corollary 4.5. *The protocol $\Pi^{\mathrm{rev}}_{\mathrm{COM}}$ implements $\mathcal{F}_{\mathrm{COM}}$ (q.v. Appendix A.1) UC-secure against a corrupted receiver. The simulation depicted in Figure 5 is indistinguishable from the real model.*

Proof. Consider the following sequence of experiments.

Experiment 1: This is the real model.

Experiment 2: The same as Experiment 1, except that the commitment sender commits to pure randomness in the commit phase and runs in the unveil phase a complete search over all token inputs to equivocate the commitment to his real input (which requires to reset the token exponentially many times).

Experiment 3: The same as Experiment 2, except that the complete search in the equivocation step is only over token inputs x on which the token functionality $x \mapsto \big(t(x), t'(x)\big)$ coincides with the mapping $x \mapsto \big(p(x), t'(x)\big)$, where p denotes the polynomial computed by Ext from the token program and the transcript of the setup phase.

Experiment 4: The ideal model, conditioned on the event that the simulator does not give up.

Experiment 5: This is the ideal model.

Experiment 1 and Experiment 2 are indistinguishable, because the commitment is statistically hiding (Lemma 4.4). Indistinguishability between Experiment 2 and Experiment 3 follows from the negligibility of Ext's failure probability ρ' (Lemma 3.1). Experiment 3 and Experiment 4 are indistinguishable by construction of the simulator—here we need that by [CS09] one finds solutions for a polynomial equation system that are statistically close to random solutions (cf. Appendix C). Experiment 4 and Experiment 5 are indistinguishable, since the simulator has polynomial expected runtime complexity (Lemma D.1) and thus gives up only with negligible probability. □

5 Multiple OT from $O(1)$ Tokens

5.1 Multiple OT with combined abort

We adapt and enhance a protocol idea by [GIMS10] for a single OT instance in the stateless-token model. It works as follows. The OT-receiver first commits to his choice bit. The OT-sender then programs a token T_{OT} and provides it with all his random coins and the message transcript of the commitment protocol. The token implements the following functionality. Upon receiving an unveil message for a bit c, the token checks if the unveil is correct; if so, it will provide an OT output s_c. The token T_{OT} is transferred to the receiver, he unveils to it his choice bit and learns the corresponding OT output.

Since the commitments of [GIMS10] in the stateless-token model require the commitment receiver to access some token in the unveil phase, they need the OT-sender to encapsulate tokens into each other. We can circumvent this by our commitment scheme $\Pi^{\mathrm{rev}}_{\mathrm{COM}}$, where the commitment

Protocol $\Pi^{\text{c-ab}}_{\text{MOT}}$

Implicitly parametrized by the number n of single OTs to be implemented. Based upon our commitment schemes $\Pi^{\text{s-o}}_{\text{COM}}$ and $\Pi^{\text{rev}}_{\text{COM}}$ and a statistically secure message authentication scheme MAC, e.g. from [WC81].

1. Sender: Let $(s_0^{(1)}, s_1^{(1)}), \ldots, (s_0^{(n)}, s_1^{(n)})$ be the sender's n OT-inputs. Sample a key k for the message authentication scheme MAC. Commit to the $2n$ values $s_0^{(i)}, s_1^{(i)}$ via $\Pi^{\text{s-o}}_{\text{COM}}$ and prepare a hardware token T_{OT} with the following functionality and send it to the receiver:
   - On input (c, w, τ, σ), verify that $c \in \{0,1\}^n$, $\sigma = \mathsf{MAC}_k(\tau)$, and w is a correct $\Pi^{\text{rev}}_{\text{COM}}$-unveil of c with commit phase transcript τ. If so, return the $\Pi^{\text{s-o}}_{\text{COM}}$-unveil messages for $s_{c_1}^{(1)}, \ldots, s_{c_n}^{(n)}$.
2. Receiver: Let $c = (c_1, \ldots, c_n)$ be the receiver's choice bits. Commit to c via $\Pi^{\text{rev}}_{\text{COM}}$.
3. Sender: Take the message transcript τ of Step 2, compute $\sigma = \mathsf{MAC}_k(\tau)$, and send σ to the receiver.
4. Receiver: Let w be the $\Pi^{\text{rev}}_{\text{COM}}$-unveil message for c. Input (c, w, τ, σ) into T_{OT}; let $(r_1, \ldots, r_n)$ denote the response. Verify that $r_1, \ldots, r_n$ are correct unveil messages for the corresponding $\Pi^{\text{s-o}}_{\text{COM}}$-commitments from Step 1 indexed by c. If so, output the unveiled values; otherwise abort.

Figure 6: Reduction of multiple OT with combined abort to our commitment protocols.

receiver does not access any tokens at all. So far, we can implement one OT instance with two tokens. Now, if we implement many OT instances in parallel the straightforward way, i.e. letting the receiver unveil all his choice bits to the token T_{OT}, we run into trouble: Each of the many OT outputs by T_{OT} can arbitrarily depend on all choice bits. Therefore, we let the sender first commit to the OT outputs via our construction $\Pi^{\text{s-o}}_{\text{COM}}$. The token T_{OT} then merely unveils the requested OT outputs. Still, T_{OT} can abort depending on all the choice bits, but we are fine with this for the moment and deal with it in the next section. Thus, our OT construction implements a flawed version of the ideal multiple-OT functionality, where a corrupted sender can additionally upload a predicate that decides whether the receiver's choice bits are accepted (cf. Appendix A.3). A similar level of security was achieved by [IKO+11] in the context of non-interactive secure computation.

There is one further refinement of the protocol, by which we achieve that all tokens can be issued independently of the parties' OT inputs. So far, the program code of T_{OT} depends on the message sent by the OT-receiver for the $\Pi^{\text{rev}}_{\text{COM}}$-commitment on his choice bits. Instead, the token sender can give the receiver an information-theoretic MAC for this message, the receiver can input it together with the unveil message into T_{OT}, and the code of T_{OT} thus needs to depend only on the MAC-key—note that by construction of $\Pi^{\text{s-o}}_{\text{COM}}$, the unveil messages that T_{OT} outputs are independent of the committed secrets. The complete OT protocol is given in Figure 6.

Lemma 5.1. *The protocol $\Pi^{\text{c-ab}}_{\text{MOT}}$ (q.v. Figure 6) UC-implements $\mathcal{F}^{\text{c-ab}}_{\text{MOT}}$ (q.v. Appendix A.3).*

Proof-sketch. We first show UC-security against a corrupted OT-receiver. In this case, the simulator can fake a real protocol run, exploiting extractability of $\Pi^{\text{rev}}_{\text{COM}}$-commitments and equivocality of $\Pi^{\text{s-o}}_{\text{COM}}$-commitments. The simulation basically works as follows. Step 1 of $\Pi^{\text{c-ab}}_{\text{MOT}}$ is simulated straightforwardly with random input for the simulated sender. In Step 2, the corrupted receiver's choice bits $(c_1, \ldots, c_n)$ can be extracted (using the sender simulator for $\Pi^{\text{rev}}_{\text{COM}}$) and sent to the ideal functionality $\mathcal{F}^{\text{c-ab}}_{\text{MOT}}$. Then, Step 3 again is simulated straightforwardly. Finally, in Step 4, the unveil messages output by the simulated token T_{OT} are replaced (using the receiver simulator for $\Pi^{\text{s-o}}_{\text{COM}}$) such that the commitments from Step 1 are equivocated to the OT-outputs $\hat{s}_{c_1}^{(1)}, \ldots, \hat{s}_{c_n}^{(n)}$ received from $\mathcal{F}^{\text{c-ab}}_{\text{MOT}}$. Indistinguishability of the simulation from the real model follows from the UC-security of $\Pi^{\text{s-o}}_{\text{COM}}$ and $\Pi^{\text{rev}}_{\text{COM}}$ and the unforgeability of the message authentication scheme MAC.

Next, we show UC-security against a corrupted OT-sender. The simulator works as follows. In Step 1 of $\Pi^{\text{c-ab}}_{\text{MOT}}$, the corrupted sender's OT inputs $(s_0^{(1)}, s_1^{(1)}), \ldots, (s_0^{(n)}, s_1^{(n)})$ can be extracted (using the sender simulator for $\Pi^{\text{s-o}}_{\text{COM}}$). Step 2 and Step 3 of $\Pi^{\text{c-ab}}_{\text{MOT}}$ are simulated straightforwardly with random input for the simulated receiver. Then the simulator has to send the extracted OT inputs $(s_0^{(1)}, s_1^{(1)}), \ldots, (s_0^{(n)}, s_1^{(n)})$ together with an abort predicate Q to the ideal functionality $\mathcal{F}^{\text{c-ab}}_{\text{MOT}}$. The predicate Q is defined by the following program, parametrized with the $\Pi^{\text{s-o}}_{\text{COM}}$-commitments and the token code T^*_{OT} obtained in Step 1 of $\Pi^{\text{c-ab}}_{\text{MOT}}$, the transcript τ of Step 2, and σ from Step 3:

1. Upon input $c \in \{0,1\}^n$, use the receiver simulator for $\Pi^{\text{rev}}_{\text{COM}}$ to obtain an unveil message $\widehat{w}$ that equivocates τ to c.
2. Run T^*_{OT} on input $(c, \widehat{w}, \tau, \sigma)$; let $(r_1, \ldots, r_n)$ denote the response.
3. Simulate the check in Step 4 of $\Pi^{\text{c-ab}}_{\text{MOT}}$, i.e., verify that $r_1, \ldots, r_n$ are correct unveil messages for the corresponding $\Pi^{\text{s-o}}_{\text{COM}}$-commitments indexed by c. If so, accept; otherwise reject.

Indistinguishability of the simulation from the real model just follows from the UC-security of $\Pi^{\text{s-o}}_{\text{COM}}$ and $\Pi^{\text{rev}}_{\text{COM}}$. □

Remark 5.2. Though stated as a three-token construction, our protocol $\Pi^{\text{c-ab}}_{\text{MOT}}$ can as well be implemented with two tokens, if one allows a token to be queried twice. In particular, the token T_{OT} gets with w a complete transcript of the messages sent to the token used in the subprotocol $\Pi^{\text{rev}}_{\text{COM}}$ anyway. Hence, even if maliciously issued tokens can keep a complex state, it does not compromise security if these two tokens are combined into one query-twice token.

5.2 How to get rid of the combined-abort flaw

The question of how to implement ideal oblivious transfer from the flawed version $\mathcal{F}^{\text{c-ab}}_{\text{MOT}}$ is closely related to the research field of OT combiners. However, an OT combiner needs access to *independent* OT instances, some of which may be corrupted. In contrast, $\mathcal{F}^{\text{c-ab}}_{\text{MOT}}$ leaks a predicate over the receiver's *joint* inputs for the multiple OT instances. Therefore we need an OT extractor, as defined in [IKOS09], rather than an OT combiner. However, the scope of [IKOS09] is skew to ours. They consider semi-honest parties, which follow the protocol, and only the leakage function is chosen maliciously. In this setting they aim at a constant extraction rate. In contrast, we consider malicious parties that may try to cheat in the extraction protocol, but we do not care much about the rate. For our purpose it suffices to implement n ideal OT instances from $n^{O(1)}$ flawed instances.

Our solution follows the basic idea of [IPS08] to take an *outer protocol* with many parties and emulate some of the parties by an *inner protocol*, such that the security features of both protocols complement each other. However, before we describe our solution, we briefly sketch why a more classic OT combiner based on 2-universal hashing would be insecure in our case. Such combiners are usually built such that the receiver's ideal-OT choice bits are basically 2-universal hash values of his flawed-OT inputs, which are uniformly random, and similarly for the outputs. In the $\mathcal{F}^{\text{c-ab}}_{\text{MOT}}$-hybrid model, such an approach is susceptible to the following generic attack. The sender just follows the protocol, except that he randomly chooses two of his $\mathcal{F}^{\text{c-ab}}_{\text{MOT}}$-input tuples, say $\left(\tilde{s}_0^{(i)}, \tilde{s}_1^{(i)}\right)$ and $\left(\tilde{s}_0^{(j)}, \tilde{s}_1^{(j)}\right)$, and flips the bits of $\tilde{s}_1^{(i)}$ and $\tilde{s}_1^{(j)}$. Furthermore, he defines the abort predicate Q such that it rejects the receiver's choice bits $\tilde{c} := (\tilde{c}_1, \tilde{c}_2, \ldots)$ if and only if $\tilde{c}_i = \tilde{c}_j = 1$. This attack has the following effect. With non-negligible probability, the 2-universal hash functions are chosen such that

- the receiver's i-th flawed-OT input-output tuple $(\tilde{c}_i, \tilde{r}_i)$ influences the calculation of an ideal-OT input-output tuple (c_k, r_k), but not (c_l, r_l), where l is an index such that
- the receiver's j-th flawed-OT input-output tuple $(\tilde{c}_j, \tilde{r}_j)$ influences (c_l, r_l), but not (c_k, r_k).

In such a case, it happens with probability $\frac{1}{2}$ that (c_k, r_k) is affected by the bit-flip of $\tilde{s}_1^{(i)}$, namely if $\tilde{c}_i = 1$. Likewise, (c_l, r_l) is affected by the bit-flip of $\tilde{s}_1^{(j)}$ if $\tilde{c}_j = 1$. Both events are statistically independent of each other, but by definition of the abort predicate Q it will never happen that the receiver produces regular output while (c_k, r_k) and (c_l, r_l) are both affected by the attack. This correlation between the joint distribution of the receiver's ideal-OT inputs and outputs and the event of an abort is not simulatable with an ideal OT.

Our construction is at an abstract level very similar to the OT combiner of [HIKN08]. We believe that the constructions of [HIKN08] can also be proven secure when based on $\mathcal{F}_{\text{MOT}}^{\text{c-ab}}$, but we prefer to present a simple combination of results from the literature as opposed to tampering with the proof details. Our final OT construction consists of three ingredients:

1. Our $\mathcal{F}_{\text{MOT}}^{\text{c-ab}}$ implementation,
2. a construction for general, statistically UC-secure two-party computation in the OT-hybrid model, e.g. from [IPS08], and
3. a statistically UC-secure protocol for multiple OT based on a single untrusted stateful tamper-proof hardware token, which we take from [DKMQ11].

We take the token functionality from [DKMQ11] and implement it by secure two-party computation from [IPS08], based on $\mathcal{F}_{\text{MOT}}^{\text{c-ab}}$ instead of ideal OT. Note that OT can be stored and reversed [Bea95, WW06] and therefore it suffices to query $\mathcal{F}_{\text{MOT}}^{\text{c-ab}}$ just once in the beginning with the token receiver being also the OT-receiver. The "emulated token" then replaces all token queries in the otherwise unchanged protocol of [DKMQ11]. Now, any specification of the abort predicate in $\mathcal{F}_{\text{MOT}}^{\text{c-ab}}$ directly corresponds to a maliciously programmed token that stops functioning depending on its inputs. Since the construction of [DKMQ11] is UC-secure against any malicious token behavior, we finally obtain UC-secure OT.

Remark 5.3. Notice that this directly provides an impossibility result for commitments in the stateless-token model where the unveil phase consists only of a single message from the sender to the receiver and local computations (without accessing any tokens) by the receiver. Otherwise our OT construction could be implemented in the stateless-token model (without encapsulation), contradicting the impossibility result for OT given in [GIMS10].

6 Bounded-Resettable Zero-Knowledge Proofs of Knowledge

We modify the constant-round zero-knowledge protocol of [GK96] for 3-COLOR such that the prover becomes resettable and only two tokens have to be sent to the verifier. In the protocol of [GK96], the verifier first commits to his challenge (the edges determining the vertices that are to be revealed), then the prover commits to permutations of the colored vertices. The verifier then reveals the challenge and the prover opens the specified commitments. The main problem imposed by a *resettable* prover is that a malicious verifier could try to run the same protocol several times, each time with different challenges, and hence step by step learn the prover's witness. The standard technique to deal with this is to let the prover's color permutations depend on the verifier's commitment in a pseudorandom way. Since we aim for *statistical* zero-knowledge, we cannot use a pseudorandom function, but need to replace it by a random polynomial of sufficient degree.

For our construction, we replace the computational commitments in [GK96] with the statistical commitments presented in the previous sections. Though, our commitment schemes have an interactive setup phase and become insecure if the token issuer is resettable. However, by fixing λ in the setup phase of $\Pi_{\text{COM}}^{\text{rev}}$, the resulting commitment scheme $\widetilde{\Pi}_{\text{COM}}^{\text{rev}}$ becomes resettable and remains statistically hiding (cf. Lemma 4.4). Making all the prover's random choices dependent on the verifier's first $\widetilde{\Pi}_{\text{COM}}^{\text{rev}}$-commit message c^{rev} is the lever we use to obtain resettability. This

Protocol $\Pi^{\text{b-r}}_{\text{SZK}}$

Implicitly parametrized by a simple 3-colorable graph $G = (V, E)$ and a query bound q in the sense that a malicious verifier can reset the prover at most $q - 1$ times. Let $n := |V|$, $t := n \cdot |E|$, and $V := \{1, \ldots, n\}$.

Auxiliary input for prover: A 3-coloring of G, denoted $\varphi : V \to \{1, 2, 3\}$.

Setup phase:

- Prover: Select a random degree-q polynomial $f \in \mathbb{F}_{2^l}[X]$, where l is the number of random bits needed for token generation in $\Pi^{\text{s-o}}_{\text{COM}}$ for $n \cdot t$ commitments. Further, select a random degree-q polynomial $g \in \mathbb{F}_{2^k}[X]$, where k is the number of random bits needed to generate t random permutations over $\{1, 2, 3\}$. W.l.o.g., l and k are larger than the commit message length in $\widetilde{\Pi}^{\text{rev}}_{\text{COM}}$. Create two tokens T^{rev} and $\mathsf{T}^{\text{s-o}}$ with the following functionalities and send them to the receiver:
  - T^{rev}: Just implement the token functionality of $\widetilde{\Pi}^{\text{rev}}_{\text{COM}}$.
  - $\mathsf{T}^{\text{s-o}}$: Upon input (x, c^{rev}), simulate the token generation procedure of $\Pi^{\text{s-o}}_{\text{COM}}$ with randomness $f(c^{\text{rev}} \| 0 \ldots 0)$, evaluate the generated token program on input x, and output the result.

Proof phase:

1. Verifier: Uniformly and independently select a random value $\lambda^{\text{s-o}}$ according to the setup phase of $\Pi^{\text{s-o}}_{\text{COM}}$ and a t-tuple of edges $\bar{E} = (\{u_1, v_1\}, \ldots, \{u_t, v_t\})$ as a challenge for the zero-knowledge proof. Use $\widetilde{\Pi}^{\text{rev}}_{\text{COM}}$ to commit to $(\bar{E}, \lambda^{\text{s-o}})$ and send the corresponding commit message c^{rev} to the prover.
2. Prover: Compute $r = f(c^{\text{rev}} \| 0 \ldots 0)$ and $r' = g(c^{\text{rev}} \| 0 \ldots 0)$. Use r' to select t random permutations $\pi_1, \ldots, \pi_t$ over $\{1, 2, 3\}$ and set $\phi_i(v) = \pi_i(\varphi(v))$ for each $v \in V$ and $i \in \{1, \ldots, t\}$. Use r to simulate the token generation of $\Pi^{\text{s-o}}_{\text{COM}}$ and compute the corresponding $\Pi^{\text{s-o}}_{\text{COM}}$-commit message $c^{\text{s-o}}$ to commit to $\phi_i(v)$ for all $v \in V$ and $i \in \{1, \ldots, t\}$. Send $c^{\text{s-o}}$ to the verifier.
3. Verifier: Send c^{rev} and the corresponding $\widetilde{\Pi}^{\text{rev}}_{\text{COM}}$-unveil message to the prover, thus unveiling $(\bar{E}, \lambda^{\text{s-o}})$.
4. Prover: If the unveil was not correct, abort. Else, compute $r = f(c^{\text{rev}} \| 0 \ldots 0)$ and simulate the token generation of $\Pi^{\text{s-o}}_{\text{COM}}$ as in Step 2. Compute the response $\tilde{p}^{\text{s-o}}$ for $\lambda^{\text{s-o}}$ according to the setup phase of $\Pi^{\text{s-o}}_{\text{COM}}$. Let $w^{\text{s-o}}$ be the $\Pi^{\text{s-o}}_{\text{COM}}$-unveil message for the commitments indexed by $\bar{E}$. Send $(\tilde{p}^{\text{s-o}}, w^{\text{s-o}})$ to the verifier.
5. Verifier: Check the unveiled commitments according to the unveil phase of $\Pi^{\text{s-o}}_{\text{COM}}$. Also verify for each edge $\{u_i, v_i\} \in \bar{E}$ that $\phi_i(u_i) \neq \phi_i(v_i)$. If all checks are passed, accept the proof; if not, reject.

Figure 7: Construction of a bounded-resettable statistical zero-knowledge proof of knowledge.

particularly has to include the randomness used in $\Pi^{\text{s-o}}_{\text{COM}}$ for token generation. Therefore, we need to use a modified token in $\Pi^{\text{s-o}}_{\text{COM}}$ with input domain $X \times C$, where C is the set of all possible commit messages c^{rev} in $\widetilde{\Pi}^{\text{rev}}_{\text{COM}}$ and X is the input space for the original token program in $\Pi^{\text{s-o}}_{\text{COM}}$. On input (x, c^{rev}), the modified $\Pi^{\text{s-o}}_{\text{COM}}$-token first simulates the token generation of $\Pi^{\text{s-o}}_{\text{COM}}$ with randomness c^{rev} and then runs the generated token program on input x. See Figure 7 for all further details.

Lemma 6.1. *The protocol $\Pi^{\text{b-r}}_{\text{SZK}}$ UC-implements $\mathcal{F}_{\text{ZK}}$ (q.v. Appendix A.4).*

Proof-sketch. We first show UC-security against a corrupted verifier, i.e., the simulator must fake a protocol run without knowing a witness. In Step 1 of $\Pi^{\text{b-r}}_{\text{SZK}}$, we exploit that $\widetilde{\Pi}^{\text{rev}}_{\text{COM}}$ is still UC-secure against a corrupted commitment sender and thus the challenge $\bar{E}$ can be extracted. Then, in Step 2, the simulated prover can commit to different colorings for each challenged vertex pair $\{u_i, v_i\} \in \bar{E}$ and to arbitrary colorings otherwise. The remaining protocol is just simulated straightforwardly. Indistinguishability from a real protocol run follows, because $\Pi^{\text{s-o}}_{\text{COM}}$ is statistically hiding.

We move on to show UC-security against a corrupted prover, i.e., the simulator has to extract a witness. The complete simulation just follows the real protocol. If in the end the simulated verifier accepts, the sender simulator for $\Pi_{\mathrm{COM}}^{\text{s-o}}$ (provided with the corresponding message transcript and the token code $\mathsf{T}^{\text{s-o}}$) is used to extract the commitments from Step 2 of $\Pi_{\mathrm{SZK}}^{\text{b-r}}$, which yields t colorings for the graph G. If none of them is a valid 3-coloring, the simulator gives up; otherwise he sends a valid one to the ideal functionality $\mathcal{F}_{\mathrm{ZK}}$. It remains to show that the simulator gives up only with negligible probability. However, if none of the committed colorings is a valid 3-coloring, then the proof is accepted by the simulated verifier at most with the following probability (abstracting from the negligible case that some commitment is successfully broken by the corrupted prover):

$$\left(1-\tfrac{1}{|E|}\right)^t \;=\; \left(1-\tfrac{1}{|E|}\right)^{n\cdot|E|} \;=\; \exp\!\left(n\cdot|E|\cdot\log\!\left(1-\tfrac{1}{|E|}\right)\right) \;\leq\; \exp\!\left(n\cdot|E|\cdot\left(-\tfrac{1}{|E|}\right)\right) \;=\; \exp(-n)\;\square$$

Remark 6.2. Furthermore, $\Pi_{\mathrm{SZK}}^{\text{b-r}}$ is bounded-resettably zero-knowledge. The resettability of the prover follows from two facts. Firstly, the prover's randomness (r, r') depends deterministically but otherwise unpredictable by the verifier on his first message c^{rev}. Secondly, by the binding property of $\widetilde{\Pi}_{\mathrm{COM}}^{\mathrm{rev}}$, a corrupted verifier cannot cheat in Step 3 of $\Pi_{\mathrm{SZK}}^{\text{b-r}}$ other than switch to another instance of the zero-knowledge protocol with unrelated prover randomness (r, r').

Remark 6.3. Our construction $\Pi_{\mathrm{SZK}}^{\text{b-r}}$ can directly be used to obtain a *non-interactive* zero-knowledge proof of knowledge scheme in the bounded-resettable hardware model by storing the prover functionality in another token (or two other tokens, if each token should be queried only once).

References

[AS92] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs; a new characterization of NP. In *Foundations of Computer Science - Proceedings of FOCS 1992*, pages 2–13. IEEE Computer Society, 1992.

[Bea95] Donald Beaver. Precomputing oblivious transfer. In Don Coppersmith, editor, *Advances in Cryptology - Proceedings of CRYPTO '95*, volume 963 of *Lecture Notes in Computer Science*, pages 97–109. Springer, 1995.

[BGGL01] Boaz Barak, Oded Goldreich, Shafi Goldwasser, and Yehuda Lindell. Resettably-sound zero-knowledge and its applications. In *Foundations of Computer Science - Proceedings of FOCS 2001*, pages 116–125. IEEE Computer Society, 2001.

[BLV06] Boaz Barak, Yehuda Lindell, and Salil P. Vadhan. Lower bounds for non-black-box zero knowledge. *J. Comput. Syst. Sci.*, 72(2):321–391, 2006.

[BP13] Nir Bitansky and Omer Paneth. On the impossibility of approximate obfuscation and applications to resettable cryptography. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing - Proceedings of STOC 2013*, pages 241–250. ACM, 2013.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Foundations of Computer Science - Proceedings of FOCS 2001*, pages 136–145. IEEE Computer Society, 2001. Revised full version online available at `http://eprint.iacr.org/2000/067`.

[CGGM00] Ran Canetti, Oded Goldreich, Shafi Goldwasser, and Silvio Micali. Resettable zero-knowledge (extended abstract). In F. Frances Yao and Eugene M. Luks, editors, *Symposium on Theory of Computing - Proceedings of STOC 2000*, pages 235–244. ACM, 2000.

[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+14] Seung Geol Choi, Jonathan Katz, Dominique Schröder, Arkady Yerukhimovich, and Hong-Sheng Zhou. (Efficient) universally composable oblivious transfer using a minimal number of stateless tokens. In Yehuda Lindell, editor, *Theory of Cryptography - Proceedings of TCC 2014*, volume 8349 of *Lecture Notes in Computer Science*, pages 638–662. Springer, 2014.

[COPV13] Kai-Min Chung, Rafail Ostrovsky, Rafael Pass, and Ivan Visconti. Simultaneous resettability from one-way functions. In *Foundations of Computer Science - Proceedings of FOCS 2013*, pages 60–69. IEEE Computer Society, 2013.

[COSV12] Chongwon Cho, Rafail Ostrovsky, Alessandra Scafuro, and Ivan Visconti. Simultaneously resettable arguments of knowledge. In Ronald Cramer, editor, *Theory of Cryptography - Proceedings of TCC 2012*, volume 7194 of *Lecture Notes in Computer Science*, pages 530–547. Springer, 2012.

[CPS13] Kai-Min Chung, Rafael Pass, and Karn Seth. Non-black-box simulation from one-way functions and applications to resettable security. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing - Proceedings of STOC 2013*, pages 231–240. ACM, 2013.

[CPV04a] Giovanni Di Crescenzo, Giuseppe Persiano, and Ivan Visconti. Constant-round resettable zero knowledge with concurrent soundness in the bare public-key model. In Matthew K. Franklin, editor, *Advances in Cryptology - Proceedings of CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 237–253. Springer, 2004.

[CPV04b] Giovanni Di Crescenzo, Giuseppe Persiano, and Ivan Visconti. Improved setup assumptions for 3-round resettable zero knowledge. In Pil Joong Lee, editor, *Advances in Cryptology - Proceedings of ASIACRYPT 2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 530–544. Springer, 2004.

[CS09] Mahdi Cheraghchi and Amin Shokrollahi. Almost-uniform sampling of points on high-dimensional algebraic varieties. In Susanne Albers and Jean-Yves Marion, editors, *Symposium on Theoretical Aspects of Computer Science - Proceedings of STACS 2009*, volume 3 of *LIPIcs*, pages 277–288. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany, 2009.

[DFG+11] Yi Deng, Dengguo Feng, Vipul Goyal, Dongdai Lin, Amit Sahai, and Moti Yung. Resettable cryptography in constant rounds - the case of zero knowledge. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - Proceedings of ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 390–406. Springer, 2011.

[DGS09] Yi Deng, Vipul Goyal, and Amit Sahai. Resolving the simultaneous resettability conjecture and a new non-black-box simulation strategy. In *Foundations of Computer Science - Proceedings of FOCS 2009*, pages 251–260. IEEE Computer Society, 2009.

[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*. Springer, 2011. Extended full version online available at `http://eprint.iacr.org/2012/135`.

[DKW11] Stefan Dziembowski, Tomasz Kazana, and Daniel Wichs. Key-evolution schemes resilient to space-bounded leakage. In Phillip Rogaway, editor, *Advances in Cryptology - Proceedings of CRYPTO 2001*, volume 6841 of *Lecture Notes in Computer Science*, pages 335–353. Springer, 2011.

[DL08] Yi Deng and Dongdai Lin. Resettable zero knowledge with concurrent soundness in the bare public-key model under standard assumption. In Dingyi Pei, Moti Yung, Dongdai Lin, and Chuankun Wu, editors, *Information Security and Cryptology - Proceedings of Inscrypt 2007*, volume 4990 of *Lecture Notes in Computer Science*, pages 123–137. Springer, 2008.

[DMMQN13] Nico Döttling, Thilo Mie, Jörn Müller-Quade, and Tobias Nilges. Implementing resettable UC-functionalities with untrusted tamper-proof hardware-tokens. In *Theory of Cryptography - Proceedings of TCC 2013*, 2013.

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.

[DS13] Ivan Damgård and Alessandra Scafuro. Unconditionally secure and universally composable commitments from physical assumptions. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - Proceedings of ASIACRYPT 2013*, volume 8270 of *Lecture Notes in Computer Science*, pages 100–119. Springer, 2013.

[FGL+91] Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. Approximating clique is almost NP-complete (preliminary version). In *Foundations of Computer Science - Proceedings of FOCS 1991*, pages 2–12. IEEE Computer Society, 1991.

[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, 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.

[GK96] Oded Goldreich and Ariel Kahan. How to construct constant-round zero-knowledge proof systems for NP. *J. Cryptology*, 9(3):167–190, 1996.

[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.

[GM11] Vipul Goyal and Hemanta K. Maji. Stateless cryptographic protocols. In Rafail Ostrovsky, editor, *Foundations of Computer Science - Proceedings of FOCS 2011*, pages 678–687. IEEE, 2011.

[GS09] Vipul Goyal and Amit Sahai. Resettably secure computation. In Antoine Joux, editor, *Advances in Cryptology - Proceedings of EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 54–71. Springer, 2009.

[HIKN08] Danny Harnik, Yuval Ishai, Eyal Kushilevitz, and Jesper Buus Nielsen. OT-combiners via secure computation. In Ran Canetti, editor, *Theory of Cryptography - Proceedings of TCC 2008*, volume 4948 of *Lecture Notes in Computer Science*, pages 393–411. Springer, 2008.

[IKO+11] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, Manoj Prabhakaran, and Amit Sahai. Efficient non-interactive secure computation. In Kenneth G. Paterson, editor, *Advances in Cryptology - Proceedings of EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 406–425. Springer, 2011.

[IKOS09] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Extracting correlations. In *Foundations of Computer Science - Proceedings of FOCS 2009*, pages 261–270. IEEE Computer Society, 2009.

[IMS12] Yuval Ishai, Mohammad Mahmoody, and Amit Sahai. On efficient zero-knowledge PCPs. In Ronald Cramer, editor, *Theory of Cryptography - Proceedings of TCC 2012*, volume 7194 of *Lecture Notes in Computer Science*, pages 151–168. Springer, 2012.

[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*, 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*. Springer, 2007.

[KP01] Joe Kilian and Erez Petrank. Concurrent and resettable zero-knowledge in polyloalgorithm rounds. In Jeffrey Scott Vitter, Paul G. Spirakis, and Mihalis Yannakakis, editors, *Symposium on Theory of Computing - Proceedings of STOC 2001*, pages 560–569. ACM, 2001.

[KPT97] Joe Kilian, Erez Petrank, and Gábor Tardos. Probabilistically checkable proofs with zero knowledge. In Frank Thomson Leighton and Peter W. Shor, editors, *Symposium on Theory of Computing - Proceedings of STOC 1997*, pages 496–505. ACM, 1997.

[KR08] Yael Tauman Kalai and Ran Raz. Interactive PCP. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages, and Programming - Proceedings of*

ICALP 2008, Part II, volume 5126 of *Lecture Notes in Computer Science*, pages 536–547. Springer, 2008.

[KV09] Jonathan Katz and Vinod Vaikuntanathan. Signature schemes with bounded leakage resilience. In Mitsuru Matsui, editor, *Advances in Cryptology - Proceedings of ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 703–720. Springer, 2009.

[MR01] Silvio Micali and Leonid Reyzin. Min-round resettable zero-knowledge in the public-key model. In Birgit Pfitzmann, editor, *Advances in Cryptology - Proceedings of EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 373–393. Springer, 2001.

[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.

[WC81] Mark N. Wegman and Larry Carter. New hash functions and their use in authentication and set equality. *J. Comput. Syst. Sci.*, 22(3):265–279, 1981.

[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*. Springer, 2006.

[YZ07] Moti Yung and Yunlei Zhao. Generic and practical resettable zero-knowledge in the bare public-key model. In Moni Naor, editor, *Advances in Cryptology - Proceedings of EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Computer Science*, pages 129–147. Springer, 2007.

[ZDLZ03] Yunlei Zhao, Xiaotie Deng, Chan H. Lee, and Hong Zhu. Resettable zero-knowledge in the weak public-key model. In Eli Biham, editor, *Advances in Cryptology - Proceedings of EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 123–139. Springer, 2003.

Appendices

A Ideal Functionalities

In this section we provide the ideal functionalities for our security proofs in the UC-framework. For better readability, we omit session identifiers and cover only the two-party case for each protocol.

A.1 Ideal functionality for a single commitment

Functionality $\mathcal{F}_{\text{COM}}$

Implicitly parametrized by a domain of secrets S.

Commit phase:

1. Await an input $(\texttt{commit}, s)$ with $s \in S$ from the sender. Then, store s and send $(\texttt{committed})$ to the adversary.
2. Await a message $(\texttt{notify})$ from the adversary. Then send $(\texttt{committed})$ to the receiver.

Unveil phase:

3. Await an input $(\texttt{unveil}, \hat{s})$ with $\hat{s} \in S$ from the sender. Then, store $\hat{s}$ and send $(\texttt{opened})$ to the adversary.
4. Await a message $(\texttt{output})$ from the adversary. Then, if $\hat{s} = s$, send $\hat{s}$ to the receiver; otherwise, send a special reject message $\perp$.

A.2 Ideal functionality for multiple commitment with selective opening

Functionality $\mathcal{F}_{\text{COM}}^{\text{s-o}}$

Implicitly parametrized by a domain of secrets S and the number n of commitments to be implemented.

Commit phase:

1. Await an input $(\texttt{commit}, s)$ with $s = (s_1, \ldots, s_n) \in S^n$ from the sender. Then, store s and send $(\texttt{committed})$ to the adversary.
2. Await a message $(\texttt{notify})$ from the adversary. Then send $(\texttt{committed})$ to the receiver.

Unveil phase:

3. Await an input $(\texttt{unveil}, I, \hat{s})$ with $I \subseteq \{1, \ldots, n\}$ and $\hat{s} = (\hat{s}_i)_{i \in I} \in S^{|I|}$ from the sender. Then, store $(I, \hat{s})$ and send $(\texttt{opened})$ to the adversary.
4. Await a message $(\texttt{output})$ from the adversary. Then, if $\hat{s} = (s_i)_{i \in I}$, send $(I, \hat{s})$ to the receiver; otherwise, send a special reject message $\perp$.

A.3 Ideal functionality for multiple oblivious transfer with combined abort

Functionality $\mathcal{F}_{\text{MOT}}^{\text{c-ab}}$

Implicitly parametrized by a sender input domain S and the number n of single OTs to be implemented.

- Upon input $\big(\texttt{create}, (s_0^{(1)}, s_1^{(1)}), \ldots, (s_0^{(n)}, s_1^{(n)})\big)$ with $(s_0^{(i)}, s_1^{(i)}) \in S \times S$ from the sender, verify that the sender is uncorrupted; otherwise ignore that input. Next, store $(s_0^{(0)}, s_1^{(0)}), \ldots, (s_0^{(n)}, s_1^{(n)})$, send $(\texttt{sent})$ to the adversary, and henceforth ignore any further input from the sender.
- Upon input $\big(\texttt{mal\_create}, (s_0^{(1)}, s_1^{(1)}), \ldots, (s_0^{(n)}, s_1^{(n)}), Q\big)$ with $(s_0^{(i)}, s_1^{(i)}) \in S \times S$ and a predicate $Q : \{0,1\}^n \to \{\texttt{accept}, \texttt{reject}\}$ from the sender, verify that the sender is corrupted; otherwise ignore that input. Next, store $(s_0^{(0)}, s_1^{(0)}), \ldots, (s_0^{(n)}, s_1^{(n)})$ and Q, send $(\texttt{sent})$ to the adversary, and henceforth ignore any further input from the sender.
- Upon input $(\texttt{choice}, c)$ with $c = (c_1, \ldots, c_n) \in \{0,1\}^n$ from the receiver, store c, send $(\texttt{chosen})$ to the adversary, and henceforth ignore any further input from the receiver.
- Upon receiving a message $(\texttt{output})$ from the adversary, check that there are stored inputs $(s_0^{(1)}, s_1^{(1)}), \ldots, (s_0^{(n)}, s_1^{(n)})$ from the sender and c from the receiver; else ignore this message. If the sender is corrupted, compute $Q(c)$ and abort if $Q(c) = \texttt{reject}$. Next, send $(s_{c_1}^{(1)}, \ldots, s_{c_n}^{(n)})$ to the receiver and ignore any further $(\texttt{output})$-messages from the adversary.
- Upon receiving a message $(\texttt{notify})$ from the adversary, check that there are stored inputs $(s_0^{(1)}, s_1^{(1)}), \ldots, (s_0^{(n)}, s_1^{(n)})$ from the sender and c from the receiver; else ignore this message. Next, send an empty output to the sender and ignore any further $(\texttt{notify})$-messages from the adversary.

A.4 Ideal functionality for zero-knowledge

Functionality $\mathcal{F}_{\text{ZK}}$

Implicitly parametrized with an *NP*-language L and a corresponding *NP*-problem instance x.

1. Await an input w from the sender. Then, store w and send $(\texttt{sent})$ to the adversary.
2. Await a message $(\texttt{verify})$ from the adversary. Then, if w is a witness for $x \in L$, send $(\texttt{accept})$ to the verifier; else send $(\texttt{reject})$.

B Graph Lemmata

B.1 Notations

Throughout Appendix B we use the following notations.

Notation (Vectors of polynomials). Let $\mathbb{F}$ be an arbitrary field. We canonically extend the notion of polynomials over $\mathbb{F}$ as follows. By $\mathbb{F}^n[X]$ we denote the set of all n-tuples $(p_1, \ldots, p_n)$ of polynomials $p_1, \ldots, p_n \in \mathbb{F}[X]$. Each polynomial $p := (p_1, \ldots, p_n) \in \mathbb{F}^n[X]$ defines a function $\mathbb{F} \to \mathbb{F}^n$, $x \mapsto \big(p_1(x), \ldots, p_n(x)\big)$. We treat $\mathbb{F}^n[X]$ as an $\mathbb{F}$-linear vector space in the natural way.

Notation (Interpolation polynomials). Let $\mathbb{F}$ be a finite field and let $n \in \mathbb{N}$. Given any mapping $f : \mathbb{F} \to \mathbb{F}^n$ and any non-empty set $M \subseteq \mathbb{F}$, let $\hat{\mathrm{p}}_M^{(f)} \in \mathbb{F}^n[X]$ denote the unique polynomial of degree at most $|M| - 1$, such that $\hat{\mathrm{p}}_M^{(f)}(x) = f(x)$ for all $x \in M$.

Notation (Generalization of the degree operator). Let $\mathbb{F}$ be a finite field and let $n \in \mathbb{N}$. For each polynomial $p := (p_1, \ldots, p_n) \in \mathbb{F}^n[X]$ we define its degree as $\deg(p) := \max_{i=1}^n(\deg(p_i))$. Further,

given any mapping $f : \mathbb{F} \to \mathbb{F}^n$ and any non-empty set $M \subseteq \mathbb{F}$, let $\deg_M(f) := \deg(\hat{\mathrm{p}}_M^{(f)})$. For convenience we set $\deg_\emptyset(f) := -\infty$ and $\deg_M(f, f') := \max\big(\deg_M(f), \deg_M(f')\big)$.

Notation (Bipartite graphs and neighborhoods). We denote a bipartite graph G as a triple (V, U, E), where V and U are the vertex sets of the two parts of G and $E \subseteq V \times U$ is the set of edges of G. The neighborhood of any vertex $v \in V \cup U$ in G is denoted as $\mathcal{N}_G(v)$, or $\mathcal{N}(v)$ for short, and its degree as $\deg_G(v) := \big|\mathcal{N}_G(v)\big|$. Note that $\mathcal{N}_G(v) \subseteq U$ for all $v \in V$ and $\mathcal{N}_G(u) \subseteq V$ for all $u \in U$.

B.2 Bounds for distinct check polynomials

Lemma B.1. *Let any finite field $\mathbb{F}$, a dimension $n \in \mathbb{N}$ and two mappings $t, t' : \mathbb{F} \to \mathbb{F}^n$ be given. Further, let some family of polynomials $\{\tilde{p}_\lambda\}_{\lambda \in \mathbb{F}} \subseteq \mathbb{F}^n[X]$ be given, let $k := \max_{\lambda \in \mathbb{F}}(\deg(\tilde{p}_\lambda))$ and for each $\lambda \in \mathbb{F}$ let $M_\lambda := \{x \in \mathbb{F} \mid \lambda \cdot t(x) + t'(x) = \tilde{p}_\lambda(x)\}$. Then it holds for all distinct $\lambda_1, \lambda_2 \in \mathbb{F}$ that $\deg_{M_{\lambda_1} \cap M_{\lambda_2}}(t, t') \leq k$.*

Proof. We first show:

$$\text{For each } M \subseteq \mathbb{F} \text{ there exists at most one } \lambda \in \mathbb{F} \text{ with } \deg_M(\lambda \cdot t + t') < \deg_M(t, t'). \tag{1}$$

Note that $\hat{\mathrm{p}}_M^{(\alpha f + \beta g)} = \alpha \cdot \hat{\mathrm{p}}_M^{(f)} + \beta \cdot \hat{\mathrm{p}}_M^{(g)}$ and thus $\deg_M(\alpha f + \beta g) \leq \max\big(\deg_M(f), \deg_M(g)\big)$ for arbitrary mappings $f, g : \mathbb{F} \to \mathbb{F}^n$ and any coefficients $\alpha, \beta \in \mathbb{F}$. Thus, if $\deg_M(\lambda_1 \cdot t + t') < \deg_M(t)$ and $\deg_M(\lambda_2 \cdot t + t') < \deg_M(t)$ for some distinct $\lambda_1, \lambda_2 \in \mathbb{F}$, we had the following contradiction:

$$\deg_M((\lambda_1 - \lambda_2) \cdot t) \;=\; \deg_M((\lambda_1 \cdot t + t') - (\lambda_2 \cdot t + t')) \;<\; \deg_M(t)$$

Analogously, if $\deg_M(\lambda_1 \cdot t + t') < \deg_M(t')$ and $\deg_M(\lambda_2 \cdot t + t') < \deg_M(t')$ for some distinct $\lambda_1, \lambda_2 \in \mathbb{F}$, we had the following contradiction:

$$\deg_M((\lambda_1 - \lambda_2) \cdot t') \;=\; \deg_M(\lambda_1 \cdot (\lambda_2 \cdot t + t') - \lambda_2 \cdot (\lambda_1 \cdot t + t')) \;<\; \deg_M(t')$$

Thus, (1) must be true and can be used to prove our lemma.

Let any distinct $\lambda_1, \lambda_2 \in \mathbb{F}$ be given. Note that $\deg_{M_{\lambda_1} \cap M_{\lambda_2}}(\lambda_1 \cdot t + t') = \deg_{M_{\lambda_1} \cap M_{\lambda_2}}(\tilde{p}_{\lambda_1}) \leq k$ by definition of M_{λ_1}, and $\deg_{M_{\lambda_1} \cap M_{\lambda_2}}(\lambda_2 \cdot t + t') = \deg_{M_{\lambda_1} \cap M_{\lambda_2}}(\tilde{p}_{\lambda_2}) \leq k$ by definition of M_{λ_2}. Thus, if $\deg_{M_{\lambda_1} \cap M_{\lambda_2}}(t, t') > k$, we had that $\deg_{M_{\lambda_1} \cap M_{\lambda_2}}(\lambda_1 \cdot t + t') < \deg_{M_{\lambda_1} \cap M_{\lambda_2}}(t, t')$ and also $\deg_{M_{\lambda_1} \cap M_{\lambda_2}}(\lambda_2 \cdot t + t') < \deg_{M_{\lambda_1} \cap M_{\lambda_2}}(t, t')$, which would contradict (1). □

Lemma B.2. *Let any finite field $\mathbb{F}$, a dimension $n \in \mathbb{N}$ and two mappings $t, t' : \mathbb{F} \to \mathbb{F}^n$ be given. Further, let some family of polynomials $\{\tilde{p}_\lambda\}_{\lambda \in \mathbb{F}} \subseteq \mathbb{F}^n[X]$ be given, let $k := \max_{\lambda \in \mathbb{F}}(\deg(\tilde{p}_\lambda))$ and for each $\lambda \in \mathbb{F}$ let $M_\lambda := \{x \in \mathbb{F} \mid \lambda \cdot t(x) + t'(x) = \tilde{p}_\lambda(x)\}$. Then for all $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \in \mathbb{F}$ with $\lambda_1 \neq \lambda_2$ and $\lambda_3 \neq \lambda_4$ and $M_{\lambda_1} \cap M_{\lambda_2} \neq M_{\lambda_3} \cap M_{\lambda_4}$ we have that $k \geq |M_{\lambda_1} \cap M_{\lambda_2} \cap M_{\lambda_3} \cap M_{\lambda_4}|$.*

Proof. Let any $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \in \mathbb{F}$ with $\lambda_1 \neq \lambda_2$ and $\lambda_3 \neq \lambda_4$ be given. Moreover, assume that $k < |M_{\lambda_1} \cap M_{\lambda_2} \cap M_{\lambda_3} \cap M_{\lambda_4}|$. We have to show that $M_{\lambda_1} \cap M_{\lambda_2} = M_{\lambda_3} \cap M_{\lambda_4}$.

Since any two polynomials $p, p' \in \mathbb{F}^n[X]$ which coincide on more than $\max\big(\deg(p), \deg(p')\big)$ nodes are identical, we observe:

1. For all $M, M' \subseteq \mathbb{F}$ with $|M \cap M'| > \max\big(\deg_M(t, t'), \deg_{M'}(t, t')\big)$ it holds that $\hat{\mathrm{p}}_M^{(t)} = \hat{\mathrm{p}}_{M'}^{(t)}$ and $\hat{\mathrm{p}}_M^{(t')} = \hat{\mathrm{p}}_{M'}^{(t')}$.

2. For all $\lambda \in \mathbb{F}$ and $M \subseteq M_\lambda$ with $\deg_M(t, t') \leq k < |M|$ we have that $\tilde{p}_\lambda = \hat{\mathrm{p}}_M^{(t)} + \lambda \cdot \hat{\mathrm{p}}_M^{(t')}$.

Now, as $\deg_{M_{\lambda_1}\cap M_{\lambda_2}}(t,t') \leq k$ and $\deg_{M_{\lambda_3}\cap M_{\lambda_4}}(t,t') \leq k$ by Lemma B.1, it follows by our Observation 1 that $\hat{\mathrm{p}}^{(t)}_{M_{\lambda_1}\cap M_{\lambda_2}} = \hat{\mathrm{p}}^{(t)}_{M_{\lambda_3}\cap M_{\lambda_4}}$ and $\hat{\mathrm{p}}^{(t')}_{M_{\lambda_1}\cap M_{\lambda_2}} = \hat{\mathrm{p}}^{(t')}_{M_{\lambda_3}\cap M_{\lambda_4}}$. Let $p := \hat{\mathrm{p}}^{(t)}_{M_{\lambda_1}\cap M_{\lambda_2}}$ and $p' := \hat{\mathrm{p}}^{(t')}_{M_{\lambda_1}\cap M_{\lambda_2}}$. Note that $\deg(p) \leq k$ and $\deg(p') \leq k$ by Lemma B.1. Putting things together, we get by our Observation 2 that $\tilde{p}_{\lambda_i} = p + \lambda_i \cdot p'$ for $i = 1, \ldots, 4$. However, as $p(x) = t(x)$ and $p'(x) = t'(x)$ for all $x \in (M_{\lambda_1} \cap M_{\lambda_2}) \cup (M_{\lambda_3} \cap M_{\lambda_4})$ by construction, it follows that $M_{\lambda_i} \supseteq (M_{\lambda_1} \cap M_{\lambda_2}) \cup (M_{\lambda_3} \cap M_{\lambda_4})$ for $i = 1, \ldots, 4$, and hence $M_{\lambda_1} \cap M_{\lambda_2} = M_{\lambda_3} \cap M_{\lambda_4}$. □

B.3 Decomposition of bipartite graphs into complete bipartite subgraphs

Lemma B.3. *Let U be some finite universe and let $\varepsilon, \delta \in \mathbb{R}$, such that $|U|^{\delta} \leq |U|^{\varepsilon}/4$. Further, let $m \in \mathbb{N}_{>0}$ and $N_1, \ldots, N_m \subseteq U$, such that $|N_i| \geq |U|^{1-\delta}$ and $|N_i \cap N_j| \leq |U|^{1-\varepsilon-\delta}$ for all distinct i, j. Then we have that $m < |U|^{\varepsilon}/2$.*

Proof. W.l.o.g. it suffices to show that $m \neq \lceil |U|^{\varepsilon}/2 \rceil$. We call $x \in U$ a *shared element*, if $x \in N_i \cap N_j$ for some distinct indices i, j. Note that each N_i contains at most $(m-1) \cdot |U|^{1-\varepsilon-\delta}$ shared elements, since by assumption $|N_i \cap N_j| \leq |U|^{1-\varepsilon-\delta}$ for all $j \neq i$. Hence, each N_i must contain at least $|U|^{1-\delta} - (m-1) \cdot |U|^{1-\varepsilon-\delta}$ non-shared elements. In other words, if $m = \lceil |U|^{\varepsilon}/2 \rceil$, we can estimate:

$$|U| \quad \geq \quad m \cdot \left(|U|^{1-\delta} - (m-1) \cdot |U|^{1-\varepsilon-\delta}\right) \quad > \quad \tfrac{|U|^{\varepsilon}}{2} \cdot \left(|U|^{1-\delta} - \tfrac{|U|^{\varepsilon}}{2} \cdot |U|^{1-\varepsilon-\delta}\right) \quad = \quad \tfrac{|U|^{1+\varepsilon-\delta}}{4}$$

However, since $|U|^{\varepsilon-\delta} \geq 4$ by assumption, this is a contradiction and thus concludes our proof. □

Lemma B.4. *Let a finite bipartite graph $G := (V, U, E)$ and some constant $k \in \mathbb{R}$ be given, such that $k \geq \left|\mathcal{N}_G(v) \cap \mathcal{N}_G(\bar{v}) \cap \mathcal{N}_G(v') \cap \mathcal{N}_G(\bar{v}')\right|$ for all vertices $v, \bar{v}, v', \bar{v}' \in V$ with $v \neq \bar{v}$ and $v' \neq \bar{v}'$ and $\mathcal{N}_G(v) \cap \mathcal{N}_G(\bar{v}) \neq \mathcal{N}_G(v') \cap \mathcal{N}_G(\bar{v}')$. Then, for any $\varepsilon, \delta \in \mathbb{R}$ with $16 \cdot |U|^{2\delta} \leq |U|^{\varepsilon} \leq \sqrt{|U|/k}$, there exists a subset of edges $E' \subseteq E$ such that $|E'| < |V| \cdot |U|^{1-\delta} + |U|^{1+\varepsilon}$ and each connected component of $G' := (V, U, E \backslash E')$ is either a single vertex or a complete bipartite graph.*

Proof. W.l.o.g. we can choose ε such that $16 \cdot |U|^{2\delta} = |U|^{\varepsilon}$, and k such that $|U|^{\varepsilon} = \sqrt{|U|/k}$. We define the auxiliary constants ε' and δ' as follows: Let $\varepsilon' := \delta + \log_{|U|} 4$, i.e. $|U|^{\delta} = |U|^{\varepsilon'}/4$, and let $\delta' := \varepsilon' + \delta$. Note that $|U|^{\delta'} = 4 \cdot |U|^{2\delta} = |U|^{\varepsilon}/4$ and hence $k = |U|^{1-2\varepsilon} = |U|^{1-\varepsilon-\delta'}/4$ by construction. Now, by the following three steps, we transform our graph G into G' by removing edges—w.l.o.g. we even remove vertices together with all their adjacent edges. We remove at most $|V| \cdot |U|^{1-\delta}$ edges in Step 1 and less than $|U|^{1+\varepsilon}/2$ edges in each of Step 2 and Step 3, which sums up to less than $|V| \cdot |U|^{1-\delta} + |U|^{1+\varepsilon}$ removed edges as claimed.

Step 1: We first remove all $v \in V$ with $|\mathcal{N}_G(v)| \leq |U|^{1-\delta}$, thus deleting at most $|V| \cdot |U|^{1-\delta}$ edges.

Step 2: Next, we remove all vertices $v \in V$ with $\max_{v' \in V \backslash \{v\}} |\mathcal{N}_G(v) \cap \mathcal{N}_G(v')| \leq |U|^{1-\varepsilon'-\delta}$, thus deleting less than $|U|^{1+\varepsilon'}/2$ edges according to Lemma B.3. Note that $|U|^{1+\varepsilon'}/2 < |U|^{1+\varepsilon}/2$, since even $|U|^{\varepsilon'+\delta} = |U|^{\varepsilon}/4$ by construction.

Step 3: Finally, we find a mapping $\sigma : V \to \mathcal{P}(U)$, where $\mathcal{P}(U)$ denotes the power set of U, such that σ assigns to each vertex $v \in V$ a maximum possible set of neighbors $N \subseteq \mathcal{N}_G(v)$ that can be written as $N = \mathcal{N}_G(v) \cap \mathcal{N}_G(\bar{v})$ with $\bar{v} \neq v$. Note that $|\sigma(v)| > |U|^{1-\delta'}$ for all $v \in V$ due to the vertex removal in Step 2. Further note that by construction $\mathcal{N}_G(v) \cap \mathcal{N}_G(v') = \sigma(v)$ for all distinct $v, v' \in V$ with $\sigma(v) = \sigma(v')$. Hence, by the upper bound nature of k it must hold that $\left|\sigma(v) \cap \sigma(v')\right| \leq k$ for all $v, v' \in V$ with $\sigma(v) \neq \sigma(v')$. Since $k = |U|^{1-\varepsilon-\delta'}/4$ by construction, it follows by Lemma B.3 that the image space of σ consists of less than $|U|^{\varepsilon}/2$ different vertex sets $N \subseteq U$.

Now we are going to remove all edges (v,u), where $u \notin \sigma(v)$. This obviously transforms G into a disjoint composition of complete bipartite graphs (plus some unconnected vertices). So, there is only left to show that there exist no more than $|U|^{1+\varepsilon}/2$ such edges.

Consider any $N \in \sigma(V)$. As already mentioned above, we have that $\mathcal{N}_G(v) \cap \mathcal{N}_G(v') = \sigma(v)$ for all distinct vertices $v, v' \in V$ with $\sigma(v) = \sigma(v')$, or in other words, $\mathcal{N}_G(v) \cap \mathcal{N}_G(v') = N$ for all distinct $v, v' \in \sigma^{-1}(N)$. Hence, no $u \in U \setminus N$ can be adjacent to distinct vertices $v, v' \in \sigma^{-1}(N)$. I.e., there cannot exist more than $|U \setminus N|$ edges $(v,u) \in E$ with $\sigma(v) = N$ and $u \notin N$. Thus, in this final step we are removing at most $\sum_{N \in \sigma(V)} |U \setminus N|$ edges. We can estimate this by $|U|^{1+\varepsilon}/2$ however, since we have already shown above that the image space of σ consists only of less than $|U|^{\varepsilon}/2$ different vertex sets $N \subseteq U$. □

B.4 Putting things together

Corollary B.5. *Let $\mathbb{F}$ be a finite field, let $n \in \mathbb{N}$, and let $G := (V,U,E)$ be a bipartite graph such that $|V| = |U| = |\mathbb{F}|$. Let each vertex in V be identifiable with a unique field element $\lambda \in \mathbb{F}$ and likewise let each vertex in U be identifiable with a unique field element $x \in \mathbb{F}$. Moreover, let a mapping $\mathsf{S}^* : V \to \mathbb{F}^n[X]$, $\lambda \mapsto \tilde{p}_\lambda$ and a mapping $\mathsf{T}^* : U \to \mathbb{F}^n \times \mathbb{F}^n$, $x \mapsto \big(t(x), t'(x)\big)$ be given, such that $(\lambda, x) \in E$ if and only if $\tilde{p}_\lambda(x) = \lambda \cdot t(x) + t'(x)$. Let $k := \max_{\lambda \in V}(\deg(\tilde{p}_\lambda))$ be polynomially bounded in $\log|\mathbb{F}|$. Then there exists a subset of edges $E' \subseteq E$ such that $|E'| \leq |\mathbb{F}|^{2-\Omega(1)}$ and for each vertex $\lambda \in V$ we have that $\deg_{\mathcal{N}_{G'}(\lambda)}(\mathsf{T}^*) \leq k$, where $G' := (V,U,E \setminus E')$.*

Proof. By Lemma B.2 we have that $k \geq \big|\mathcal{N}_G(\lambda_1) \cap \mathcal{N}_G(\lambda_2) \cap \mathcal{N}_G(\lambda_3) \cap \mathcal{N}_G(\lambda_4)\big|$ for all vertices $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \in V$ with $\lambda_1 \neq \lambda_2$ and $\lambda_3 \neq \lambda_4$ and $\mathcal{N}_G(\lambda_1) \cap \mathcal{N}_G(\lambda_2) \neq \mathcal{N}_G(\lambda_3) \cap \mathcal{N}_G(\lambda_4)$. Now we pick some arbitrary constants $\varepsilon, \delta \in \mathbb{R}_{>0}$ with $2\delta < \varepsilon < \frac{1}{2}$. Assuming a sufficiently large field $\mathbb{F}$, we get that $16 \cdot |U|^{2\delta} \leq |U|^{\varepsilon} \leq \sqrt{|U|/k}$. This allows us to apply Lemma B.4, whereby we find a subset of edges $E' \subseteq E$ such that $|E'| < |\mathbb{F}|^{2-\delta} + |\mathbb{F}|^{1+\varepsilon}$ and each connected component of $G' := (V,U,E \setminus E')$ is either a single vertex or a complete bipartite graph. However, this turns out not sufficient to conclude our proof; we also have to get rid of all vertices $x \in U$ with $\deg_{G'}(x) = 1$. So, we also remove the corresponding edges from G', ending up with a graph $G'' := (V,U,E \setminus E'')$ where $|E''| < |\mathbb{F}|^{2-\delta} + |\mathbb{F}|^{1+\varepsilon} + |\mathbb{F}|$.

Now we show that $\deg_{\mathcal{N}_{G''}(\lambda)}(\mathsf{T}^*) \leq k$ for all $\lambda \in V$. Let $\lambda_1 \in V$ be arbitrary but fixed. W.l.o.g., $\mathcal{N}_{G''}(\lambda_1) \neq \emptyset$. Since there are no vertices $x \in U$ with $\deg_{G''}(x) = 1$ any more, we find some $\lambda_2 \in V \setminus \{\lambda_1\}$ with $\mathcal{N}_{G''}(\lambda_1) \cap \mathcal{N}_{G''}(\lambda_2) \neq \emptyset$. Since each connected component of G'' still is either a single vertex or a complete bipartite graph, it even holds that $\mathcal{N}_{G''}(\lambda_1) = \mathcal{N}_{G''}(\lambda_2)$ and by Lemma B.1 it follows that $\deg_{\mathcal{N}_{G''}(\lambda_1)}(\mathsf{T}^*) = \deg_{\mathcal{N}_{G''}(\lambda_2)}(\mathsf{T}^*) = \deg_{\mathcal{N}_{G''}(\lambda_1) \cap \mathcal{N}_{G''}(\lambda_2)}(\mathsf{T}^*) \leq k$. □

C Sampling Uniformly from Varieties of Constant Codimension

We briefly state the main theorem of [CS09], which is used in the proof of Corollary 4.5.

Theorem C.1 ([CS09, Theorem 1.1])**.** *Let $k > 0$ be a constant integer, $n > k$ and $d > 0$ be integers, let p^ℓ be a sufficiently large prime power and $\epsilon > 0$ be an arbitrarily small constant. Suppose that $f_1, \ldots, f_k \in \mathbb{F}_{p^\ell}[x_1, \ldots, x_n]$ are polynomials, each of total degree at most d, and let*

$$V \;=\; V(f_1, \ldots, f_k) \;=\; \{\xi \in \mathbb{F}_{p^\ell}^n \mid f_1(\xi) = \ldots = f_k(\xi) = 0\}$$

be the variety defined by $f_1, \ldots, f_k$. There exists a randomized algorithm that, given the description of $f_1, \ldots, f_k$ as a list of their nonzero monomials, outputs a random point $v \in \mathbb{F}_{p^\ell}^n$ such that the

distribution of v is $\frac{6}{p^{\ell(1-\epsilon)}}$-close to the uniform distribution on V. The worst-case runtime complexity of this algorithm is polynomial in $n, d, \ell \log(p)$ and the description of $f_1, \ldots, f_k$.

Concretely, in Corollary 4.5 the field is $\mathbb{F}_{2^\ell}$ and $n = 4$, as elements of $\mathbb{F}_{2^{4\ell}}$ are interpreted as 4-dimensional vectors over $\mathbb{F}_{2^\ell}$. The variety V is given by the polynomials (in $x = (x_1, \ldots, x_4)$)

$$\begin{aligned} p_1(x) - \tilde{y} &= 0 \\ \langle h \,|\, x \rangle - m + \hat{s} &= 0 \end{aligned}$$

where $p_1(x) - \tilde{y} \in \mathbb{F}_{2^\ell}[x_1, \ldots, x_4]$ is a polynomial of degree q and $\langle h \,|\, x \rangle - m + \hat{s} \in \mathbb{F}_{2^\ell}[x_1, \ldots, x_4]$ is trivially a polynomial of degree 1. Thus the parameters are $k = 2$, $n = 4$, $p = 2$, and $d = q$. We can set $\epsilon = \frac{1}{2}$ and Theorem C.1 yields an efficient algorithm that samples $\frac{6}{2^{\ell/2}}$-close to uniform from V.

D Expected polynomial runtime of the receiver simulator for $\Pi^{\text{rev}}_{\text{COM}}$

Lemma D.1. *The simulator in Figure 5 (Section 4.2) has expected polynomial runtime complexity.*

Proof. There are two steps in the simulation that might take super-polynomial time. One is the execution of Ext and the other one is the repeated sampling of $\hat{x}$. We first show that the execution of Ext is polynomially efficient on average. It suffices to bound the expected number of queries from Ext to T^*, as all other calculations by Ext are efficient by Lemma 3.1. However, again by Lemma 3.1, we have that Ext queries T^* only $(q+1)/\rho$ times on average, where ρ is exactly the probability that the protocol is not aborted in the commit phase. Thus, the expected number of queries from Ext to T^* is just $q + 1$.

Now we estimate the expected number of resamplings of $\hat{x}$. In the following let M denote the set of token inputs that pass the consistency check and let N denote the subset of token inputs for which the extracted polynomial p is correct, i.e., $M = \{x \in \mathbb{F}_{2^{4\ell}} \,|\, \lambda \cdot t(x) + t'(x) = \tilde{p}(x)\}$ and $N = \{x \in M \,|\, t(x) = p(x)\}$. Let ρ' be the failure probability from Lemma 3.1. Since $\rho' \leq 2^{-\Omega(\ell)}$, we find some $\varepsilon \in \mathbb{R}_{>0}$ with $\rho' \leq 2^{-\varepsilon\ell}$, assuming a sufficiently large security parameter ℓ. Note that $\rho' = \mathrm{E}\big(|M \setminus N| \cdot |\mathbb{F}_{2^{4\ell}}|^{-1}\big)$ just by construction. We distinguish the following three cases.

1. It happens that $|M| \leq 2^{(4-\varepsilon/4)\ell}$. However, conditioned on any choice of λ we just have that $|M| \cdot |\mathbb{F}_{2^{4\ell}}|^{-1}$ is the probability of a non-aborted commit phase. Thus, even if the iteration bound of $2^{\sqrt{\ell}}$ is always reached, this case contributes only $2^{\sqrt{\ell} - \varepsilon\ell/4}$ to the expected number of resamplings of $\hat{x}$, which is negligible.

2. It happens that $|M \setminus N| \geq 2^{(4-3\varepsilon/4)\ell}$. Since $E\big(|M \setminus N|\big) = \rho' \cdot |\mathbb{F}_{2^{4\ell}}| \leq 2^{(4-\varepsilon)\ell}$, this may be the case at most with probability $2^{-\varepsilon\ell/4}$. Thus, even if the iteration bound of $2^{\sqrt{\ell}}$ is always reached, this case also contributes only $2^{\sqrt{\ell} - \varepsilon\ell/4}$ to the average number of resamplings of $\hat{x}$.

3. It happens that $|M| > 2^{(4-\varepsilon/4)\ell}$ and $|M \setminus N| < 2^{(4-3\varepsilon/4)\ell}$. We may consider the simulated commitment sender's token input x as uniformly random over M, because $x \notin M$ means an abort in the commit phase anyway. Each sampling of $\hat{x}$ is uniformly random (up to a negligible statistical distance δ) over the set $(\sigma_1 \circ p)^{-1}((\sigma_1 \circ t)(x)) \cap \{\hat{x} \in \mathbb{F}_{2^{4\ell}} \,|\, \langle h \,|\, \sigma(\hat{x}) \rangle = m - \hat{s}\}$. The distance δ stems from the fact that we only know how to sample *almost* uniformly from large varieties. As discussed in Appendix C, we have $\delta \leq \frac{6}{2^{\ell/2}}$. Note that m and hence also $m - \hat{s}$ is uniformly random and independent of (λ, x, h). It follows by Appendix E (Corollary E.3,

instantiated with $g := \sigma_1 \circ p$ and $f := \sigma_1 \circ t$) that each sampling of $\hat{x}$ has the following success probability:

$$\Pr[\hat{x} \in N] \;>\; \frac{|N|}{|\mathbb{F}_{2^{4\ell}}|} - \frac{|M \backslash N|}{|M|} - \frac{|\mathbb{F}_{2^\ell}|}{\sqrt{|M|}} - \delta \;=\; \frac{|M|}{|\mathbb{F}_{2^{4\ell}}|} - \frac{|M \backslash N|}{|\mathbb{F}_{2^{4\ell}}|} - \frac{|M \backslash N|}{|M|} - \frac{|\mathbb{F}_{2^\ell}|}{\sqrt{|M|}} - \delta$$

Since we assumed $|M| > 2^{(4-\varepsilon/4)\ell}$ and $|M \backslash N| < 2^{(4-3\varepsilon/4)\ell}$, we can estimate:

$$\begin{aligned}
\frac{|M \backslash N|}{|\mathbb{F}_{2^{4\ell}}|} \;&<\; 2^{-3\varepsilon\ell/4} \;<\; \frac{2^{-\varepsilon\ell/2} \cdot |M|}{|\mathbb{F}_{2^{4\ell}}|} \\
\frac{|M \backslash N|}{|M|} \;&<\; 2^{-\varepsilon\ell/2} \;<\; \frac{2^{-\varepsilon\ell/4} \cdot |M|}{|\mathbb{F}_{2^{4\ell}}|} \\
\frac{|\mathbb{F}_{2^\ell}|}{\sqrt{|M|}} \;&<\; 2^{(\varepsilon/8-1)\ell} \;<\; \frac{2^{(3\varepsilon/8-1)\ell} \cdot |M|}{|\mathbb{F}_{2^{4\ell}}|} \;<\; \frac{2^{-5\ell/8} \cdot |M|}{|\mathbb{F}_{2^{4\ell}}|} \quad \text{(w.l.o.g., } \varepsilon < 1\text{)} \\
\delta \;&\leq\; 6 \cdot 2^{-\ell/2} \;<\; \frac{6 \cdot 2^{(\varepsilon/4-1/2)\ell} \cdot |M|}{|\mathbb{F}_{2^{4\ell}}|} \;<\; \frac{6 \cdot 2^{-\ell/4} \cdot |M|}{|\mathbb{F}_{2^{4\ell}}|} \quad \text{(w.l.o.g., } \varepsilon < 1\text{)}
\end{aligned}$$

Thus, the success probability for each sampling of $\hat{x}$ is lower bounded by:

$$\frac{(1 - 2^{-\varepsilon\ell/2} - 2^{-\varepsilon\ell/4} - 2^{-5\ell/8} - 6 \cdot 2^{-\ell/4}) \cdot |M|}{|\mathbb{F}_{2^{4\ell}}|}$$

In other words, for large enough ℓ the success probability is arbitrarily close to $|M| \cdot |\mathbb{F}_{2^{4\ell}}|^{-1}$. However, this is also exactly the success probability for the consistency check in the protocol's commit phase. Let $\tilde{\rho} := |M| \cdot |\mathbb{F}_{2^{4\ell}}|^{-1}$ and $\vartheta := 1 - 2^{-\varepsilon\ell/2} - 2^{-\varepsilon\ell/4} - 2^{-5\ell/8} - 6 \cdot 2^{-\ell/4}$. Hence we can upper bound the expected number of sampling steps for $\hat{x}$ by the following term:

$$\tilde{\rho} \cdot \Big(\sum\nolimits_{j=1}^{\infty} j \cdot (1 - \vartheta \cdot \tilde{\rho})^{i-1} \cdot \vartheta \cdot \tilde{\rho} \Big) \;=\; \tilde{\rho} \cdot \frac{1}{\vartheta \cdot \tilde{\rho}} \;=\; \frac{1}{\vartheta}$$

Note that by assumption $|M| > 2^{(4-\varepsilon/4)\ell}$ and therefore $\tilde{\rho} \neq 0$.

We have shown: The expected number of sampling steps for $\hat{x}$ is upper bounded by $1 + o(1)$. □

E 2-Universal Hashing Lemmata

We use the following notations throughout Appendix E.

Notation (Random variables). We denote random variables by bold characters, e.g., $\mathbf{x}$.

Notation (Uniform distribution). By $\mathbf{x} \xleftarrow{\text{r}} X$ we denote that $\mathbf{x}$ is uniformly distributed over X.

Notation (Statistical distance). We denote the statistical distance between two random variables $\mathbf{x}, \mathbf{y}$ by $\Delta(\mathbf{x}, \mathbf{y}) := \frac{1}{2} \sum_\alpha \big| \Pr[\mathbf{x} = \alpha] - \Pr[\mathbf{y} = \alpha] \big|$.

Lemma E.1 (Generalized Leftover Hash Lemma [DORS08])**.** *Let any finite sets X, Y, Z and a tuple of random variables $(\hat{\mathbf{x}}, \mathbf{z})$ with arbitrary joint distribution over $X \times Z$ be given. Let $\mathcal{H}$ be a family of 2-universal hash functions $h : X \to Y$, and let $\mathbf{h} \xleftarrow{\text{r}} \mathcal{H}$ and $\mathbf{u} \xleftarrow{\text{r}} Y$. Then it holds:*

$$\Delta\big((\mathbf{h}(\hat{\mathbf{x}}), \mathbf{h}, \mathbf{z}) , (\mathbf{u}, \mathbf{h}, \mathbf{z}) \big) \;\leq\; \tfrac{1}{2} \sqrt{\max\nolimits_{e : Z \to X} \Pr[\hat{\mathbf{x}} = e(\mathbf{z})] \cdot |Y|}$$

Proof. This lemma is just a reformulation of [DORS08, Lemma 2.4]. □

Lemma E.2. *Let any finite sets U, M, Z with $\emptyset \neq M \subseteq U$ and an arbitrary mapping $f : U \to Z$ be given. Further, let $\mathbf{x} \xleftarrow{\text{r}} M$ and $\hat{\mathbf{x}} \xleftarrow{\text{r}} f^{-1}(f(\mathbf{x}))$, i.e., $\hat{\mathbf{x}}$ is a uniformly random f-preimage (in U) of $f(\mathbf{x})$. Then,* $\Pr[\hat{\mathbf{x}} \in M] \geq \frac{|M|}{|U|}$.

Proof. First note that for any $\beta, A, B \in \mathbb{R}$ with $B > \beta > 0$ the function $\alpha \mapsto \frac{\alpha^2}{\beta} + \frac{(A-\alpha)^2}{B-\beta}$ has a global minimum at $\alpha = \frac{\beta A}{B}$. Thus, given any finite sequences $(\alpha_z)_{z \in Z} \subset \mathbb{R}$ and $(\beta_z)_{z \in Z} \subset \mathbb{R}_{>0}$, it follows by induction on $|Z|$ that $\sum_{z \in Z} \frac{\alpha_z^2}{\beta_z} \geq \sum_{t \in Z} \frac{\alpha_z A}{B} = \frac{A^2}{B}$, where $A := \sum_{z \in Z} \alpha_z$ and $B := \sum_{z \in Z} \beta_z$.

Now for each $z \in Z$, let $U_z := f^{-1}(z)$ and $M_z := U_z \cap M$. Let $Z' := f(U)$. By our considerations above we have:

$$\sum_{x \in M} \frac{|M_{f(x)}|}{|U_{f(x)}|} \;=\; \sum_{z \in Z'} \sum_{x \in M_z} \frac{|M_z|}{|U_z|} \;=\; \sum_{z \in Z'} \frac{|M_z|^2}{|U_z|} \;\geq\; \frac{|M|^2}{|U|}$$

Hence we can conclude:

$$\Pr[\hat{\mathbf{x}} \in M] \;=\; \sum_{x \in M} \underbrace{\Pr[\mathbf{x} = x]}_{= |M|^{-1}} \cdot \underbrace{\Pr[\hat{\mathbf{x}} \in M \mid \mathbf{x} = x]}_{= |M_{f(x)}| \cdot |U_{f(x)}|^{-1}} \;\geq\; \tfrac{|M|}{|U|}$$ □

Corollary E.3. *Let any finite sets U, M, N, Z with $\emptyset \neq N \subseteq M \subseteq U$ and two mappings $f, g : U \to Z$ be given with $f(x) = g(x)$ for all $x \in N$. Further, let $\mathcal{H}$ be a family of 2-universal hash functions $h : U \to Z$. Finally, let $\mathbf{x} \xleftarrow{\text{r}} M$, $\mathbf{h} \xleftarrow{\text{r}} \mathcal{H}$, $\mathbf{m} \xleftarrow{\text{r}} Z$, $\mathbf{z} := f(\mathbf{x})$, and $\hat{\mathbf{x}} \xleftarrow{\text{r}} g^{-1}(\mathbf{z}) \cap \mathbf{h}^{-1}(\mathbf{m})$ with the convention that $\hat{\mathbf{x}} = \perp \notin U$ in case of $g^{-1}(\mathbf{z}) \cap \mathbf{h}^{-1}(\mathbf{m}) = \emptyset$. Then,* $\Pr[\hat{\mathbf{x}} \in N] > \frac{|N|}{|U|} - \frac{|M \setminus N|}{|M|} - \frac{|Z|}{\sqrt{|M|}}$.

Proof. We define the following auxiliary random variables:

$$\begin{aligned} \mathbf{z}' &:= g(\mathbf{x}) & \hat{\mathbf{x}}' &\xleftarrow{\text{r}} g^{-1}(\mathbf{z}') \cap \mathbf{h}^{-1}(\mathbf{m}) \quad \text{with } \hat{\mathbf{x}}' := \perp, \text{ if } g^{-1}(\mathbf{z}') \cap \mathbf{h}^{-1}(\mathbf{m}) = \emptyset \\ \mathbf{m}' &:= \mathbf{h}(\mathbf{x}) & \hat{\mathbf{x}}'' &\xleftarrow{\text{r}} g^{-1}(\mathbf{z}') \cap \mathbf{h}^{-1}(\mathbf{m}') \end{aligned}$$

Note that $\Delta(\mathbf{z}, \mathbf{z}') \leq \Delta\big((\mathbf{z}, \mathbf{x}), (\mathbf{z}', \mathbf{x})\big) = \Pr[f(\mathbf{x}) \neq g(\mathbf{x})] \leq \Pr[\mathbf{x} \notin N]$ and therefore, as $\mathbf{h}$ and $\mathbf{m}$ are just additional independent randomness, also $\Delta\big((\mathbf{m}, \mathbf{h}, \mathbf{z}), (\mathbf{m}, \mathbf{h}, \mathbf{z}')\big) \leq \Pr[\mathbf{x} \notin N]$. Thus we have:

$$\Pr[\hat{\mathbf{x}} \in N] \;\geq\; \Pr[\hat{\mathbf{x}}' \in N] - \Pr[\mathbf{x} \notin N] \;=\; \Pr[\hat{\mathbf{x}}' \in N] - \tfrac{|M \setminus N|}{|M|}$$

Next, note that by Lemma E.1 it holds:

$$\Delta\big((\mathbf{m}, \mathbf{h}, \mathbf{z}'), (\mathbf{m}', \mathbf{h}, \mathbf{z}')\big) \;\leq\; \tfrac{1}{2}\sqrt{\max_{e : Z \to M} \Pr[\mathbf{x} = e(\mathbf{z}')] \cdot |Z|}$$

However, for any mapping $e : Z \to M$ we can estimate $\Pr[\mathbf{x} = e(\mathbf{z}')]$ as follows:

$$\Pr[\mathbf{x} = e(\mathbf{z}')] \;=\; \sum_{z \in Z} \underbrace{\Pr[\mathbf{z}' = z]}_{\leq \frac{|M \cap g^{-1}(z)|}{|M|}} \cdot \underbrace{\Pr[\mathbf{x} = e(z) \mid \mathbf{x} \in g^{-1}(z)]}_{\leq \frac{1}{|M \cap g^{-1}(z)|}} \;\leq\; \tfrac{|Z|}{|M|}$$

Hence, $\Delta\big((\mathbf{m}, \mathbf{h}, \mathbf{z}'), (\mathbf{m}', \mathbf{h}, \mathbf{z}')\big) < \frac{|Z|}{\sqrt{|M|}}$ and we can conclude:

$$\Pr[\hat{\mathbf{x}}' \in N] \;>\; \Pr[\hat{\mathbf{x}}'' \in N] - \tfrac{|Z|}{\sqrt{|M|}} \qquad \text{and therefore} \qquad \Pr[\hat{\mathbf{x}} \in N] \;>\; \Pr[\hat{\mathbf{x}}'' \in N] - \tfrac{|M \setminus N|}{|M|} - \tfrac{|Z|}{\sqrt{|M|}}$$

So, we finally need to estimate $\Pr[\hat{\mathbf{x}}'' \in N]$. Note that $\hat{\mathbf{x}}'' \xleftarrow{\text{r}} g^{-1}(g(\mathbf{x})) \cap \mathbf{h}^{-1}(\mathbf{h}(\mathbf{x}))$ by construction, i.e., $\hat{\mathbf{x}}''$ is a uniformly random preimage of $\big(g(\mathbf{x}), \mathbf{h}(\mathbf{x})\big)$ under the mapping $x \mapsto \big(g(x), \mathbf{h}(x)\big)$. Hence it follows by Lemma E.2 that $\Pr[\hat{\mathbf{x}}'' \in M] \geq \frac{|M|}{|U|}$. However, conditioned on the event that $\hat{\mathbf{x}}'' \in M$, we can consider $\hat{\mathbf{x}}''$ just as a resampled version of $\mathbf{x}$, which is uniformly distributed. Thus, $\Pr[\hat{\mathbf{x}}'' \in N \mid \hat{\mathbf{x}}'' \in M] = \frac{|N|}{|M|}$ and therefore $\Pr[\hat{\mathbf{x}}'' \in N] = \Pr[\hat{\mathbf{x}}'' \in M] \cdot \Pr[\hat{\mathbf{x}}'' \in N \mid \hat{\mathbf{x}}'' \in M] \geq \frac{|N|}{|U|}$. □