A Modular Treatment of Blind Signatures from Identification Schemes

Eduard Hauck, Eike Kiltz, and Julian Loss

Ruhr University Bochum
{eduard.hauck,eike.kiltz,julian.loss}@rub.de

Abstract. We propose a modular security treatment of blind signatures derived from linear identification schemes in the random oracle model. To this end, we present a general framework that captures several well known schemes from the literature and allows to prove their security.
Our modular security reduction introduces a new security notion for identification schemes called One-More-Man In the Middle Security which we show equivalent to the classical One-More-Unforgeability notion for blind signatures.
We also propose a generalized version of the Forking Lemma due to Bellare and Neven (CCS 2006) and show how it can be used to greatly improve the understandability of the classical security proofs for blind signatures schemes by Pointcheval and Stern (Journal of Cryptology 2000).

Keywords. Blind Signatures

1 Introduction

Blind Signatures are a fundamental cryptographic building block. Informally, a blind signature scheme is an interactive protocol between a signer and an user in which the signer issues signatures on messages chosen by the user. There are two security requirements: *blindness* ensures that the signer cannot link a signature to the run of the protocol in which it was created and *one-more unforgeability* that the user cannot forge a new signature. Originally proposed by Chaum [12] as the basis of his e-cash system, blind signatures have since found numerous applications including e-voting [22] and anonymous credentials [13,19,9,11,10,5,3]. Despite a flurry of schemes having been published over the past three and a half decades, only a handful of works has considered blind signature schemes which are mutually efficient, instantiable from standard assumptions, and remain secure even when executed in an arbitrarily concurrent fashion. The notoriously difficult task of constructing such schemes was first tackled by Pointcheval and Stern [21]. Their groundbreaking work introduces the well-known *forking lemma* and shows how it can be applied to prove security of the Okamoto-Schnorr blind signature scheme [18] under the discrete logarithm assumption in the random oracle model (ROM) [8]. Their proof technique was subsequently employed to prove the security of further schemes [20,23,4]. Unfortunately, due to the complexity and subtlety of the argument in [21], these works present either only proof sketches [20] or follow the proof of [21] almost verbatim.

Name	Type	Definition of linear function $\mathsf{F} : \mathcal{D} \to \mathcal{R}$		$\mathcal{S}$	Collision resistance
OS	Group	$\mathsf{F} : \mathbb{Z}_q^2 \to \mathbb{G}$,	$(x_1, x_2) \mapsto g_1^{x_1} g_2^{x_2}$	$\mathbb{Z}_q$	DLOG
OGQ	RSA	$\mathsf{F} : \mathbb{Z}_\lambda \times \mathbb{Z}_N^* \to \mathbb{Z}_N^*$,	$(x_1, x_2) \mapsto a^{x_1} x_2^\lambda$	$\mathbb{Z}_\lambda$	RSA
FS	RSA	$\mathsf{F} : (\mathbb{Z}_N^*)^k \to (\mathbb{Z}_N^*)^k$,	$(x_1, \ldots, x_k) \mapsto (x_1^2, \ldots, x_k^2)$	$\mathbb{Z}_2^k$	$\mathsf{FACTORING}$

Table 1. Examples of linear function families. Group type functions are defined over $\mathbb{G}$ of prime order q with generators g_1, g_2, RSA type functions are defined over an RSA modulus $N = pq$ and $a \in \mathbb{Z}_N^*$ satisfying $\mathsf{ord}(a) > 2\lambda$. Set $\mathcal{S}$ is the challenge set.

1.1 Our Contribution: A Modular Framework for Blind Signatures

In this work, we propose a general framework which shows how to derive a blind signature scheme from any *linear function family* (with certain properties), as recently introduced by Backendahl et al. [2]. Whereas blindness can be proved directly, one-more unforgeability is proved in two modular steps. In the first step, one builds a linear identification scheme from the linear function family. One-more unforgeability of the blind signature scheme in the random oracle model is shown to be tightly equivalent to a new and natural security notion of the linear identification scheme, which we call *one-more man-in-the-middle* security. In the second, technically involved, step it is shown that the latter is implied by collision resistance of the linear function family. Our framework captures several important schemes from the literature including the Okamoto-Schnorr (OS) [18], the Okamoto-GQ (OGQ) [18], and (a slightly modified version of) the Fiat-Shamir (FS) [20] blind signature schemes and offers, for the first time, a complete and formal proof for some of them. We now provide some details of our contributions.

Linear Function Families and Identification Schemes. A canonical identification scheme ID [1] is a three-move protocol of a specific form in which a prover P convinces a verifier Ver (holding a public key pk) that he knows the corresponding secret key sk. $\mathsf{ID} = \mathsf{ID}[\mathsf{LF}]$ is a linear identification scheme [2] if it follows a certain homomorphic structure induced by a linear function LF. For our purpose of building blind signatures, we will also require LF to be perfectly correct, collision resistant, and the kernel to contain a torsion-free element. (Note that this also makes LF many-to-one.) Example instantiations of (collision resistant) linear function families can be derived from OS, OGQ, and FS, cf. Table 1.

We introduce a natural new security notion for (arbitrary, not necessarily linear) canonical identification schemes called *One-More Man-in-the-Middle* (**OMMIM**)-security. Informally, ID is **OMMIM**-secure if it is infeasible to complete $Q_\mathsf{P} + 1$ (or more) runs of ID in the role of prover P after completing at most Q_P runs of ID in the role of verifier Ver. Note that **OMMIM** is weaker than standard Man-in-the-Middle security [15] (which we show to be unachievable for linear identification schemes) but stronger than impersonation against active attacks [14,7].

OMMIM Security of Linear Identification Schemes. Our first main result can be stated as follows:

Theorem 1 (informal). If LF is collision resistant, then $\mathsf{ID}[\mathsf{LF}]$ is **OMMIM** secure.

Our proof is based on a new Subset Forking Lemma that generalizes the one by Bellare and Neven [6] and contains many technical ingredients from [21] who prove the security of the Okamoto-Schnorr Blind Signature scheme. Unfortunately, the security bound from Theorem 1 is only meaningful if $Q_{\mathsf{V}}^{Q_{\mathsf{P}}+1} \leq |\mathcal{C}| =: q$, where Q_{V} refers to the (potentially large) number of sessions with the verifier and challenge set $\mathcal{C}$ is a parameter of the identification scheme. We next show in Theorem 2 that a natural generalization of Schnorr's ROS-problem [24] to linear functions can be used to break the **OMMIM** of $\mathsf{ID}[\mathsf{LF}]$. The ROS-problem (for the relevant parameters) becomes information theoretically hard when $Q_{\mathsf{V}}^{Q_{\mathsf{P}}+1} \leq q$. For all other cases, it can be solved in sub-exponential time $(Q_{\mathsf{V}}+1)\,2^{\sqrt{\log q}/(1+\log(Q_{\mathsf{V}}+1))}$ using Wagner's k-List algorithm [25]. Our ROS-based attack works whenever $\mathcal{C}$ is a finite field, which is the case for OS and OGQ.

Canonical Blind Signature Schemes. We introduce the notion of *canonical blind signature schemes* (BS), which are three-move blind signature schemes of a specific form. In terms of security we define *blindness* and *one-more unforgeability* (**OMUF**). Intuitively, **OMUF** states that the adversary cannot produce more valid message-signatures pairs, then it has completed successful sessions with the signer. (Note that each such session yields a valid message-signature pair.) Here we consider a natural and strong version of **OMUF** in which abandoned session with the signer (i.e., sessions that are started but never completed) are not counted as a successful sessions with the signer as they do not yield a valid message-signature pair. We propose a general compiler to convert any linear identification scheme $\mathsf{ID}[\mathsf{LF}]$ and a hash function H into a canonical blind signature scheme $\mathsf{BS}[\mathsf{LF},\mathsf{H}]$. Our second main result can be stated as follows:

Theorem 3 (informal). **OMUF** security of $\mathsf{BS}[\mathsf{LF},\mathsf{H}]$ is tightly equivalent to **OMMIM** security of $\mathsf{ID}[\mathsf{LF}]$ in the random oracle model.

Theorem 4 (informal). $\mathsf{BS}[\mathsf{LF},\mathsf{H}]$ is perfectly blind.

Figure 1.1 summarizes our modular security analysis of $\mathsf{BS}[\mathsf{LF},\mathsf{H}]$. Combining our main theorems, we obtain security proofs for the OS, OGQ, and FS blind signature schemes. Here, the number of random oracle queries Q_{H} corresponds to the number Q_{V} of open sessions with the verifier, whereas the number Q_{S} of signing sessions corresponds to the number of sessions Q_{P} with the prover. Hence, **OMUF** security of $\mathsf{BS}[\mathsf{LF},\mathsf{H}]$ is only guaranteed if $Q_{\mathsf{H}}^{Q_{\mathsf{S}}+1} \ll q$, i.e., for polylogarithmically parallel signing sessions Q_{S}. Our ROS-based attack demonstrates that this restriction is required.

1.2 Technical details

We now give an intuition for the proof of Theorem 1. Roughly, it states that one can reduce the **OMMIM** security of $\mathsf{ID}[\mathsf{LF}]$ from the problem of finding a

$$\begin{tikzcd}
\mathbf{CR}_{\mathsf{LF}} \arrow[r, "\text{Th. 1}"] & \mathbf{OMMIM}_{\mathsf{ID[LF]}} \arrow[dl, "\text{Th. 2}"'] \arrow[r, shift right] & \mathbf{OMUF}_{\mathsf{BS[LF,H]}} \arrow[l, shift right, "\text{Th. 3}"'] \\
\mathbf{ROS}_{\mathsf{LF}} & &
\end{tikzcd}$$

Fig. 1. Overview of our modular security analysis for $\mathsf{BS}[\mathsf{LF},\mathsf{H}]$. The arrows denote security implications.

non-trivial collision with respect to the linear function LF. Our proof follows the ideas of Pointcheval and Stern [21], but uses as a key ingredient a novel forking lemma, which enables us to present the proof in [21] in a much more clean and general fashion. The main idea behind our reduction is to run the adversary M against **OMMIM**-security twice, where the instance I and randomness ω in the second run are kept the same, and part of the oracle answers, denoted $\boldsymbol{h}, \boldsymbol{h'}$, are re-sampled uniformly. In this way, we hope to obtain from M two distinct values $\hat{\chi}, \hat{\chi}'$ which yield a collision with respect to LF. The main challenge in our setting is that $\hat{\chi}$ and $\hat{\chi}'$ depend on the internal state of M. To show that $\hat{\chi} \neq \hat{\chi}'$ with high probability, one requires an intricate argument that heavily builds upon a generalized version of Bellare and Neven's Forking Lemma [6]. Our lemma is tailored toward the ideas of the proof in [21] and allows for a more fine-grained replay strategy than the version of [6]. More precisely, our version of the forking lemma considers not only the probability of successfully running an algorithm twice with the same instance I, randomness ω, and (partially distinct) oracle answers $\boldsymbol{h}, \boldsymbol{h'}$, but also allows to analyze in more detail the properties of the triples $(I, \omega, \boldsymbol{h}), (I, \omega, \boldsymbol{h'})$.

1.3 Blind Signatures from Lattices?

We remark that our proof requires linear functions with perfect correctness. This leaves open the question of whether our framework can be extended to cover also the lattice-based identification scheme due to Lyubashevsky [16] and the resulting blind signature scheme due to Rückert [23]. At a technical level, imperfect correctness causes a problem in the proof of Theorem 3 which relates the **OMMIM**-security of $\mathsf{ID}[\mathsf{LF}]$ to **OMUF**-security of $\mathsf{BS}[\mathsf{LF},\mathsf{H}]$. If the adversary manages to abort even a single run of $\mathsf{BS}[\mathsf{LF},\mathsf{H}]$ in the simulated **OMUF** experiment, our reduction fails at simulating the necessary amount of completed runs of $\mathsf{BS}[\mathsf{LF},\mathsf{H}]$ to the adversary. This subtlety in the proof arises from the fact that in the **OMMIM** experiment, there is no way of telling whether a run of $\mathsf{ID}[\mathsf{LF}]$ with the adversary in the role of the verifier was completed. On the other hand, in $\mathsf{BS}[\mathsf{LF},\mathsf{H}]$, the user can prove to the signer that it obtained an invalid signature for a particular run of the protocol and hence force a restart. We leave it as an open problem to adapt our framework to linear functions with correctness errors.

2 Preliminaries and Notation

Sets and Algorithms. We denote as $h \xleftarrow{\$} \mathcal{H}$ the uniform sampling of the variable s from the set $\mathcal{H}$. If ℓ is an integer, then $[\ell]$ is the set $\{1, \ldots, \ell\}$. We write bold lower case letters $\boldsymbol{h}$ to denote a vector of elements and denote the length of $\boldsymbol{h}$ as $|\boldsymbol{h}|$. For $j > 1$, we write $\boldsymbol{h}_{[j]}$ to refer to the first j entries of $\boldsymbol{h}$. For $1 \leq j \leq Q$ and $\boldsymbol{g} \in \mathcal{H}^{j-1}$ we now define the conditional distribution $\boldsymbol{h'} \xleftarrow{\$} \mathcal{C}^{Q_\mathsf{V}}|\boldsymbol{g}$ which samples $\boldsymbol{h'} \xleftarrow{\$} \mathcal{H}^Q$ conditioned on $\boldsymbol{h'}_{[j-1]} = \boldsymbol{g}$. (This can be implemented by copying vector $\boldsymbol{g}$ into the first $j-1$ entries of $\boldsymbol{h'}$ and next sampling the subvector $\boldsymbol{h}'_j, \ldots, \boldsymbol{h}'_Q \xleftarrow{\$} \mathcal{H}^{Q-j+1}$.)

We write bold upper case letters $\mathbf{A}$ to denote matrices. We denote the i-th row vector of $\mathbf{A}$ as $\mathbf{A}_i$ and the j-th entry of $\mathbf{A}_i$ as $\mathbf{A}_{i,j}$. We use uppercase letters A, B to denote algorithms. Unless otherwise stated, all our algorithms are probabilistic and we write $(y_1, ...) \xleftarrow{\$} \mathsf{A}(x_1, ...)$ to denote that A returns $(y_1, ...)$ when run on input $(x_1, ...)$. We write A^B to denote that A has oracle access to B during its execution. Any probabilistic algorithm $\mathsf{A}(x)$, on some input x can be written as a deterministic algorithm $\mathsf{A}(x; \omega)$ on input x and randomness ω. We use standard code-based security games and write $\mathbf{G}^\mathsf{A} \Rightarrow 1$ to denote the event that algorithm A is successful in game $\mathbf{G}$.

3 Linear Functions and Identification Schemes

A *module* is specified by two sets $\mathcal{S}$ and $\mathcal{M}$, where $\mathcal{S}$ is a ring with multiplicative identity element $1_\mathcal{S}$ and $\langle \mathcal{M}, +, 0 \rangle$ is an additive Abelian group and a mapping $\cdot :$ $\mathcal{S} \times \mathcal{M} \to \mathcal{M}$, s.t. for all $r, s \in \mathcal{S}$ and $x, y \in \mathcal{M}$ we have (i) $r \cdot (x+y) = r \cdot x + r \cdot y$; (ii) $(r+s) \cdot x = r \cdot x + s \cdot x$; (iii) $(rs) \cdot x = r \cdot (s \cdot x)$; and (iv) $1_S \cdot x = x$.

Syntax of Linear Function Families. A *linear function family* LF [2] is a tuple of algorithms $(\mathsf{PGen}, \mathsf{F})$. On input the security parameter, the randomized algorithm PGen returns some parameters *par*, which implicitly define the sets $\mathcal{S} = \mathcal{S}(par), \mathcal{D} = \mathcal{D}(par)$ and $\mathcal{R} = \mathcal{R}(par)$. $\mathcal{S}$ is a set of scalars such that $\mathcal{D}$ and $\mathcal{R}$ are modules over $\mathcal{S}$. Further, $\mathsf{F}(par, \cdot)$ implements a mapping from $\mathcal{D}$ to $\mathcal{R}$. To simplify our presentation, we will omit *par* from F's input from now on. $\mathsf{F}(\cdot)$ is required to be a *module homomorphism*, meaning that for any $(x, y) \in (\mathcal{D} \times \mathcal{D})$ and $s \in \mathcal{S}$:

$$\mathsf{F}(s \cdot x + y) = s \cdot \mathsf{F}(x) + \mathsf{F}(y).$$

We say that LF has a *torsion-free element from the kernel* if for all *par* generated with PGen, there exist $z^* \in \mathcal{D} \setminus \{0\}$ such that (i) $\mathsf{F}(z^*) = 0$; and (ii) for all $s \in \mathcal{S}$ satisfying $s \cdot z^* = 0$ we have $s = 0$. Note that this implies that F is a many-to-one mapping.

Security Properties of Linear Function Families. We now define two security properties of a linear function family (collision resistance and ROS security) which will play a significant role in the subsequent sections.

We define the advantage of an adversary A, breaking the *collision resistance* of LF as

$$\mathbf{Adv}^{\mathbf{CR}}_{\mathsf{LF}}(\mathsf{A}) := \Pr_{par \xleftarrow{\$} \mathsf{PGen}, (x_1,x_2) \xleftarrow{\$} \mathsf{A}(par)} [\mathsf{F}(x_1) = \mathsf{F}(x_2) \wedge x_1 \neq x_2]$$

and say that LF is (ε, t)-**CR** if for all adversaries A running in time $\mathbf{Time}(\mathsf{A}) \leq t$ we have $\mathbf{Adv}^{\mathbf{CR}}_{\mathsf{LF}}(\mathsf{A}) \leq \varepsilon$.

The ROS (Random inhomogenities in an Overdetermined, Solvable system of linear equations) problem was introduced by Schnorr [24] (also in the context of blind signatures). Here, we generalize Schnorr's formulation to linear function families. For a linear function family LF we define the advantage of an adversary A as

$$\mathbf{Adv}^{\mathbf{ROS}}_{\mathsf{LF}}(\mathsf{A}) := \Pr[\mathbf{ROS}^{\mathsf{A}}_{\mathsf{LF}} \Rightarrow 1],$$

where game $\mathbf{ROS}_{\mathsf{LF}}$ is defined in Figure 2. We furthermore say that LF is $(\varepsilon, t, \ell, Q_{\mathsf{H}})$-**ROS** secure if for all adversaries A running in time $\mathbf{Time}(\mathsf{A}) \leq t$ and making at most Q_{H} queries to the random oracle, we have $\mathbf{Adv}^{\mathbf{ROS}}_{\mathsf{LF}}(\mathsf{A}) \leq \varepsilon$.

GAME $\mathbf{ROS}_{\mathsf{LF}}$:
$par \xleftarrow{\$} \mathsf{PGen}$
$(\boldsymbol{c} \in \mathcal{S}^{\ell+1}, \mathbf{A} \in \mathcal{S}^{(\ell+1)\times(\ell+1)}) \leftarrow \mathsf{A}^{\mathsf{H}}(par)$
If $(\boldsymbol{c}_{\ell+1} = -1) \wedge (\mathbf{A}\boldsymbol{c} = 0) \wedge (\forall i,j \in [\ell+1] : \mathsf{H}(\mathbf{A}_{i,1}, \ldots, \mathbf{A}_{i,\ell}) = \mathbf{A}_{i,\ell+1}) \wedge (\mathbf{A}_i \neq \mathbf{A}_j)$ Then
  Return 1
Return 0

Fig. 2. Game $\mathbf{ROS}_{\mathsf{LF}}$, where $\mathsf{H}\colon \{0,1\}^* \to \mathcal{S}$ is a random oracle.

The following Lemma summarizes the known hardness results for the Generalized ROS-Problem for the specific case in which $\mathcal{S}$ is a field of prime order q.

Lemma 1 ([24,25,17]). *Let* LF *be a linear function family for which* $\mathcal{S}$ *is a field of prime order* q*. For every* t*,* LF *is* $(t, \varepsilon = Q_{\mathsf{H}}^{\ell+1}/q, \ell, Q_{\mathsf{H}})$-**ROS** *secure. Conversely,* LF *is not* $(t, 1/4, \ell, Q_{\mathsf{H}})$-**ROS** *secure for* $Q_{\mathsf{H}} = (\ell+1)\, 2^{\sqrt{\log q}/(1+\log(\ell+1))}$ *and* $t = O\left((\ell+1)2^{\sqrt{\log q}/(1+\log(\ell+1))}\right)$.

Examples of Linear Function Families. We now give three examples of LF with the required properties. We remark that [2] contains more examples of linear functions, but not all of them have a torsion-free element from the kernel.

Okamoto-Schnorr. PGen returns the parameters $par := (\mathbb{G}, g_1, g_2) \xleftarrow{\$} \mathsf{PGen}(1^\lambda)$, where $g_1, g_2 \in \mathbb{G}$, q is prime, and $|\mathbb{G}| = q$. par defines sets $\mathcal{S}, \mathcal{D}, \mathcal{R}$, and the homomorphic evaluation function F as

$$\mathcal{S} := \mathbb{Z}_q; \quad \mathcal{D} := \mathbb{Z}_q^2; \quad \mathcal{R} := \mathbb{G}; \quad \mathsf{F} : \mathbb{Z}_q^2 \to \mathbb{G}, (x_1, x_2) \mapsto g_1^{x_1} g_2^{x_2}.$$

It is not hard to see that F is an homomorphism. It is also not hard to see that collision resistance of LF is equivalent to the discrete logarithm problem over $\mathbb{G}$, i.e., $\mathbf{Adv}^{\mathbf{CR}}_{\mathsf{LF}}(\mathsf{A}) = \mathbf{Adv}^{\mathsf{DLOG}}_{\mathbb{G}}(\mathsf{B})$. For all parameters *par* and for $w = \log_{g_1}(g_2)$, the element $z^* = (z_1^*, z_2^*) := (w, -1)$, yields a torsion-free in the kernel of LF since $\mathsf{F}(z^*) = g_1^w g_2^{-1} = 1$, where $1 = 0_{\mathbb{G}}$ is the neutral element in $\mathbb{G}$. Furthermore, for all $s \in \mathbb{Z}_q$ satisfying $s \cdot z^* = (s \cdot w, -s) = (0, 0)$ we have $s = 0 \bmod q$ since q is prime.

Okamoto-Guillou-Quisquater. PGen returns the parameters $par := (N = pq, \lambda, a) \xleftarrow{\$} \mathsf{PGen}(1^\lambda)$, where p, q are prime and λ is prime and co-prime with $N, \varphi(N)$ and $a \in \mathbb{Z}_N^*, \mathsf{ord}(a) > 2\lambda$. The parameters *par* define

$$\mathcal{S} := \mathbb{Z}_\lambda; \quad \mathcal{R} := \mathbb{Z}_N^*; \quad \mathcal{D} = \{(x_1, x_2 = za^{\lfloor \frac{x_1}{\lambda} \rfloor}) \bmod N \mid x_1 \in \mathbb{Z}_\lambda, z \in \mathbb{Z}_N^*\},$$

where $\mathcal{D}$ is an abelian group with the group operation $(x_1, x_2) \circ (y_1, y_2) = (x_1 + y_1 \mod \lambda, x_2 y_2 a^{\lfloor \frac{x_1+y_1}{\lambda} \rfloor} \bmod N)$. The evaluation function F is defined as

$$\mathsf{F}\colon \mathbb{Z}_\lambda \times \mathbb{Z}_N^* \to \mathbb{Z}_N^*, \mathsf{F}(x_1, x_2) := a^{x_1} x_2^\lambda.$$

F is an homomorphism, since:

$$\begin{aligned}
\mathsf{F}((x_1, x_2) \circ (y_1, y_2)) &= \mathsf{F}(x_1 + y_1 \bmod \lambda, x_2 y_2 a^{\lfloor \frac{x_1+y_1}{\lambda} \rfloor} \bmod N) \\
&= a^{x_1+y_1 \bmod \lambda} \left(x_2 y_2 a^{\lfloor \frac{x_1+y_1}{\lambda} \rfloor} \right)^\lambda \\
&= a^{((x_1+y_1) \bmod \lambda) + \lambda \lfloor \frac{x_1+y_1}{\lambda} \rfloor} (x_2 y_2)^\lambda \qquad (1) \\
&= a^{x_1+y_1} (x_2 y_2)^\lambda \qquad (2) \\
&= \mathsf{F}(x_1, x_2) \mathsf{F}(y_1, y_2),
\end{aligned}$$

where (1) and (2) follow from the identity: $(x \bmod \lambda) = x - \lambda \lfloor \frac{x}{\lambda} \rfloor$.

A collision $(x_1, x_2) \neq (y_1, y_2)$ with $\mathsf{F}(x_1, x_2) = \mathsf{F}(y_1, y_2)$ implies $a^{x_1 - y_1} = (y_2/x_2)^\lambda$ with $\gcd(\lambda, x_1 - x_2) = 1$ from which one can extract the $a^{1/\lambda}$ using the extended Euclidean Algorithm. Hence, collision resistance is implied by the RSA assumption.

For all parameters *par*, $z^* = (z_1^*, z_2^*) := (-1, a^{1/\lambda})$ is a torsion-free element in the kernel of F since $\mathsf{F}(z^*) = a^{-1 \mod \lambda} (a^{1/\lambda})^\lambda a^{\lfloor \frac{-1}{\lambda} \rfloor} = a^{(-1 \mod \lambda) + \lfloor \frac{-1}{\lambda} \rfloor} a = 1$, where $1 = 0_{\mathcal{R}}$ is the neutral element in $\mathcal{R}$. Furthermore, for all $s \in \mathbb{Z}_\lambda$ satisfying $s \cdot z^* = (-s, (a^{1/\lambda})^s a^{\lfloor \frac{-s}{\lambda} \rfloor}) = (0, 1)$ we have $s = 0 \bmod \lambda$.

Fiat-Shamir. PGen returns parameters $par := (N = pq, k)$, where p, q are prime and k is an integer. Parameters *par* define

$$\begin{aligned}
&\mathcal{S} := \mathbb{Z}_2^k; \quad \mathcal{D} := (\mathbb{Z}_N^*)^k, \mathcal{R} := (\mathbb{Z}_N^*)^k; \\
&\mathsf{F} : (\mathbb{Z}_N^*)^k \to (\mathbb{Z}_N^*)^k, \mathsf{F}(x_1, \ldots, x_k) \mapsto (x_1^2, \ldots, x_k^2).
\end{aligned}$$

Clearly, collision resistance of LF is equivalent to factorization. For all parameters *par*, $z^* = (z_1^*, \ldots, z_k^*) := (-1, \ldots, -1)$ is a torsion-free element from the kernel of F since $\mathsf{F}(z^*) = (1, \ldots, 1)$, where $(1, \ldots, 1) = 0_{\mathcal{R}}$ is the neutral element in $\mathcal{R}$. Furthermore, for all $\boldsymbol{s} \in \mathbb{Z}_2^k$ satisfying $\boldsymbol{s} \cdot z^* = (-1^{s_1}, \ldots, -1^{s_k}) = (1, \ldots, 1)$ we have $\boldsymbol{s} = \mathbf{0} \bmod 2$.

4 Canonical Identification Schemes

4.1 Syntax and Security

We now recall the definition of define canonical identification schemes [1] and discuss their security notions.

Definition 1 (Canonical Identification Scheme). *A* canonical identification scheme *is a tuple of algorithms* $\mathsf{ID} = (\mathsf{IGen}, \mathsf{P}, \mathsf{Ver})$.

- *The* key generation algorithm IGen *takes as input parameters par and outputs a public/secret key pair* (pk, sk). *We assume that pk implicitly defines a challenge set* $\mathcal{C} = \mathcal{C}(pk)$.
- *The prover algorithm* P *is split into two randomized algorithms* $\mathsf{P}_1, \mathsf{P}_2$*, i.e.,* $\mathsf{P} = (\mathsf{P}_1, \mathsf{P}_2)$. P_1 *takes as input a secret key sk and returns a commitment* R *and a state st. The deterministic algorithm* P_2 *takes as input a state st, a secret key sk, a commitment* R*, and a challenge* $c \in \mathcal{C}$. *It returns a response* s.
- *The deterministic verification algorithm* Ver *takes as input a public key pk, a commitment* R*, a challenge* $c \in \mathcal{C}$*, and a response* s. *It returns* $b \in \{0, 1\}$.

The diagram below depicts an interaction between prover P *and verifier* V. *For correctness we require that for all* $(pk, sk) \in \mathsf{IGen}(par)$*, all* $(st, R) \in \mathsf{P}_1(sk)$*, all* $c \in \mathcal{C}$*, and all* $s \in \mathsf{P}_2(sk, R, c, st)$*, it holds that* $\mathsf{Ver}(pk, R, c, s) = 1$.

Prover $\mathsf{P}(sk)$		Verifier $\mathsf{V}(pk)$
$(st, R) \xleftarrow{\$} \mathsf{P}_1(sk)$	$\xrightarrow{R}$	
	$\xleftarrow{c}$	$c \xleftarrow{\$} \mathcal{C}$
$s \leftarrow \mathsf{P}_2(sk, R, c, st)$	$\xrightarrow{s}$	$b \leftarrow \mathsf{Ver}(pk, R, c, s)$
Output 1		Output b

Standard security notions for canonical identification schemes include impersonation security against passive and active attacks, and Man-in-the-Middle security [1,7]. We now introduce a new security notion called *One-More Man-in-the-Middle* security. The One-More Man-in-the-Middle (**OMMIM**) security experiment for an identification scheme ID and an adversary A is defined in Figure 3. Adversary A simultaneously plays against a prover (modeled through oracles $\mathtt{P}_1$ and $\mathtt{P}_2$) and a verifier (modeled through oracles $\mathtt{V}_1$ and $\mathtt{V}_2$). Session identifiers *pSid* and *vSid* are used to model an interaction with the prover and the verifier, respectively. A call to $\mathtt{P}_1$ returns a new prover session identifier *pSid* and sets flag $\mathbf{pSess}_{pSid}$ to `open`. A call to $\mathtt{P}_2(pSid, \cdot)$ with the same *pSid* sets the flag $\mathbf{pSess}_{pSid}$ to `closed`. Similarly, a call to $\mathtt{V}_1$ returns a new verifier session identifier *vSid* and sets flag $\mathbf{vSess}_{vSid}$ to `open`. A call to $\mathtt{V}_2(vSid, \cdot)$ with the same *pSid* sets the flag $\mathbf{vSess}_{vSid}$ to `closed`. A closed verifier session *vSid* is successful if the oracle $\mathtt{V}_2(vSid, \cdot)$ returns 1. Lines 03-06 define several internal random variables for later references. Variable $Q_{\mathtt{P}_2}(\mathsf{A})$ counts the number of closed prover sessions and $Q_{\mathsf{P}_1}(\mathsf{A})$ counts the number of abandoned sessions (i.e., sessions that were opened but never closed). Most importantly, variable $\ell(\mathsf{A})$ counts the number of successful verifier sessions and variable $Q_{\mathsf{P}_2}(\mathsf{A})$ counts the number of closed sessions with the prover. Adversary A wins the **OMMIM** game, if $\ell(\mathsf{A}) \geq Q_{\mathsf{P}_2}(\mathsf{A}) + 1$, i.e., if A convinces the verifier in at least one more successful verifier sessions than there exist closed sessions with the prover. The **OMMIM** advantage function of an adversary A against ID is defined as $\mathbf{Adv}^{\mathbf{OMMIM}}_{\mathsf{ID}}(\mathsf{A}) := \Pr[\mathbf{OMMIM}^{\mathsf{A}}_{\mathsf{ID}} \Rightarrow 1]$.

We say that ID is $(\varepsilon, t, Q_{\mathsf{V}}, Q_{\mathsf{P}_1}, Q_{\mathsf{P}_2})$-**OMMIM** secure if for all adversaries A satisfying $\mathbf{Time}(\mathsf{A}) \leq t$, $Q_{\mathsf{V}}(\mathsf{A}) \leq Q_{\mathsf{V}}$, $Q_{\mathsf{P}_2}(\mathsf{A}) \leq Q_{\mathsf{P}_2}$, and $Q_{\mathsf{P}_1}(\mathsf{A}) \leq Q_{\mathsf{P}_1}$, we have $\mathbf{Adv}^{\mathbf{OMMIM}}_{\mathsf{ID}}(\mathsf{A}) \leq \varepsilon$.

GAME $\mathbf{OMMIM}^{\mathsf{A}}_{\mathsf{ID}}$:
$(sk, pk) \leftarrow \mathsf{IGen}$
$pSid \leftarrow 0, vSid \leftarrow 0$ //initialize prover/verifier session id
$\mathsf{A}^{\mathtt{P}_1,\mathtt{P}_2,\mathtt{V}_1,\mathtt{V}_2}(pk)$
$Q_{\mathsf{V}}(\mathsf{A}) \leftarrow vSid$ //#total sessions with verifier
$Q_{\mathsf{P}_1}(\mathsf{A}) \leftarrow \#\{1 \leq k \leq pSid \mid \mathbf{pSess}_k = \mathtt{open}\}$ //#abandoned prover sessions
$Q_{\mathsf{P}_2}(\mathsf{A}) \leftarrow \#\{1 \leq k \leq pSid \mid \mathbf{pSess}_k = \mathtt{closed}\}$ //#closed prover sessions
$\ell(\mathsf{A}) \leftarrow \#\{1 \leq k \leq vSid \mid \mathbf{vSess}_k = \mathtt{closed} \wedge \boldsymbol{b}'_k = 1\}$ //#successful verifier sessions
If $\ell(\mathsf{A}) \geq Q_{\mathsf{P}_2}(\mathsf{A}) + 1$ Then //A's winning condition
  Return 1
Return 0

Procedure $\mathtt{P}_1$
$pSid \leftarrow pSid + 1$
$\mathbf{pSess}_{pSid} \leftarrow \mathtt{open}$
$(\boldsymbol{st}_{pSid}, \boldsymbol{R}_{pSid}) \xleftarrow{\$} \mathsf{P}_1$
Return $(pSid, \boldsymbol{R}_{pSid})$

Procedure $\mathtt{P}_2(pSid, c)$
If $\mathbf{pSess}_{pSid}$ Then
  Return $\perp$
$\mathbf{pSess}_{pSid} \leftarrow \mathtt{closed}$
$\boldsymbol{s}_{pSid} \leftarrow \mathsf{P}_2(\boldsymbol{st}_{pSid}, sk, \boldsymbol{R}_{pSid}, c)$
Return $\boldsymbol{s}_{pSid}$

Procedure $\mathtt{V}_1(R')$
$vSid \leftarrow vSid + 1$
$\mathbf{vSess}_{vSid} \leftarrow \mathtt{open}$
$\boldsymbol{R}'_{vSid} \leftarrow R'; \boldsymbol{c}'_{vSid} \xleftarrow{\$} \mathcal{C}$
Return $(vSid, \boldsymbol{c}'_{vSid})$

Procedure $\mathtt{V}_2(vSid, s')$
If $\mathbf{vSess}_{vSid} \neq \mathtt{open}$ Then
  Return $\perp$
$\mathbf{vSess}_{vSid} \leftarrow \mathtt{closed}$
$\boldsymbol{b}'_{vSid} \leftarrow \mathsf{Ver}(pk, \boldsymbol{R}'_{vSid}, \boldsymbol{c}'_{vSid}, s')$
Return $\boldsymbol{b}'_{vSid}$

Fig. 3. The One-More Man-in-the-Middle security game $\mathbf{OMMIM}^{\mathsf{A}}_{\mathsf{ID}}$

We remark that impersonation against active and passive attacks is a weaker notion than **OMMIM** security, whereas Man-in-the-Middle (**MIM**) security is stronger. Concretely, in the standard **MIM** experiment the winning condition is relaxed in the sense that there only has to exist a successful session with the verifier with a transcript that does not result from a closed session with the prover.

4.2 Identification schemes from linear function families

As showed in [2], a linear function family LF directly implies a canonical identification scheme $\mathsf{ID}[\mathsf{LF}]$. The construction is given in Figure 4, where $par \overset{\$}{\leftarrow} \mathsf{PGen}$ are fixed global system parameters. We will prove later that $\mathsf{ID}[\mathsf{LF}]$ is **OMMIM** secure. This is the best we can hope for since by the linearity of LF, $\mathsf{ID}[\mathsf{LF}]$ can never be (fully) **MIM** secure. (Concretely, an adversary receiving a commitment R from the prover can send $R' = \mathsf{F}(\hat{r}) + R$ for some $\hat{r} \neq 0$ to the verifier. After forwarding $c' = c$ from verifier to prover, it receives s from the prover and submits $s' = s + \hat{r}$ to the verifier. Since $(R, c, s) \neq (R', c', s')$, A wins the **MIM** experiment with advantage 1.)

Algorithm $\mathsf{IGen}(par)$	Algorithm $\mathsf{P}_1(sk)$
$sk \overset{\$}{\leftarrow} \mathcal{D}$	$r \overset{\$}{\leftarrow} \mathcal{D}$; $R \leftarrow \mathsf{F}(r)$
$pk \leftarrow \mathsf{F}(sk)$	$st_{\mathsf{P}} := r$
Return (sk, pk)	Return (st_{P}, R)
Algorithm $\mathsf{Ver}(pk, R, c, s)$	Algorithm $\mathsf{P}_2(sk, st_{\mathsf{P}}, c)$
$S \leftarrow \mathsf{F}(s)$	$r \leftarrow st_{\mathsf{P}}$
If $S = c \cdot pk + R$ Then	$s \leftarrow c \cdot sk + r$
Return 1	Return s
Return 0	

Fig. 4. Construction of $\mathsf{ID}[\mathsf{LF}] := (\mathsf{IGen}, \mathsf{P} := (\mathsf{P}_1, \mathsf{P}_2), \mathsf{Ver})$ with challenge set $\mathcal{C} = \mathcal{S}$.

Theorem 1. *Suppose* LF *is a linear function family with a torsion-free element from the kernel. If* LF *is* (ε', t')-**CR** *secure, then* $\mathsf{ID}[\mathsf{LF}]$ *is* $(\varepsilon, t, Q_{\mathsf{V}}, Q_{\mathsf{P}_2}, Q_{\mathsf{P}_1})$-**OMMIM** *secure where*

$$t' = 2t, \quad \varepsilon' = O\left(\left(\varepsilon - \frac{(Q_{\mathsf{V}} Q_{\mathsf{P}})^{Q_{\mathsf{P}_2}+1}}{q}\right) \frac{1}{Q_{\mathsf{V}}^2 Q_{\mathsf{P}_2}^3}\right)$$

and $Q_{\mathsf{P}} = Q_{\mathsf{P}_1} + Q_{\mathsf{P}_2}$.

The proof of this theorem will be given in Section 6.

Theorem 2. *Let* LF *be a linear function family. If* $\mathsf{ID}[\mathsf{LF}]$ *is* $(\varepsilon, t, Q_{\mathsf{V}}, Q_{\mathsf{P}_2}, Q_{\mathsf{P}_1} = 0)$-**OMMIM** *secure then* LF *is* $(\varepsilon, t, \ell = Q_{\mathsf{P}_2}, Q_{\mathsf{H}} = Q_{\mathsf{V}})$-**ROS** *secure.*

Proof. Let A be an $(\varepsilon, t, \ell, Q_{\mathsf{H}})$-adversary in game **ROS**. We assume w.l.o.g. that A only makes distinct queries to the random oracle H. In Figure 5, we show how to construct an $(\varepsilon, t, Q_{\mathsf{V}}, Q_{\mathsf{P}_2}, Q_{\mathsf{P}_1})$-adversary B that is executed in game $\mathbf{OMMIM}_{\mathsf{ID}}$ and uses A as a subroutine. First, B starts Q_{P_2} sessions with the Prover oracle P_1, receiving commitments $\boldsymbol{R}$. Next, A is executed, where B answers a query of the form $\mathsf{H}(\boldsymbol{a})$ from A as $\boldsymbol{c}'_{\boldsymbol{a}}$, where $\boldsymbol{c}'_{\boldsymbol{a}} := \mathsf{V}_1(\sum_{j=1}^{Q_{\mathsf{P}_2}} \boldsymbol{a}_j \boldsymbol{R}_j)$. Note that

Adversary $\mathsf{B}^{\mathtt{P}_1,\mathtt{P}_2,\mathtt{V}_1,\mathtt{V}_2}(pk)$:
For $j \in [Q_{\mathsf{P}_2}]$ Do:
  $(\boldsymbol{pSid}_j, \boldsymbol{R}_j) \xleftarrow{\$} \mathtt{P}_1$ //start Q_{P_2} sessions with Prover
$(\boldsymbol{c} \in \mathcal{S}^{Q_{\mathsf{P}_2}+1}, \mathbf{A} \in \mathcal{S}^{(Q_{\mathsf{P}_2}+1)\times(Q_{\mathsf{P}_2}+1)}) \xleftarrow{\$} \mathsf{A}^{\mathsf{H}}(par)$
Parse $(\mathbf{Z} \in \mathcal{S}^{(Q_{\mathsf{P}_2}+1)\times Q_{\mathsf{P}_2}}, \boldsymbol{z} \in \mathcal{S}^{Q_{\mathsf{P}_2}+1}) \leftarrow \mathbf{A}$
For $j \in [Q_{\mathsf{P}_2}]$ Do:
  $\boldsymbol{s}_j \leftarrow \mathtt{P}_2(\boldsymbol{pSid}_j, \boldsymbol{c}_j)$ //close Q_{P_2} Prover sessions
For $i \in [Q_{\mathsf{P}_2}+1]$ Do:
  $\boldsymbol{s}'_i \leftarrow \sum_{j=1}^{Q_{\mathsf{P}_2}} \mathbf{A}_{i,j}\boldsymbol{s}_j$
  $\boldsymbol{b}_i \leftarrow \mathtt{V}_2(\boldsymbol{vSid}_{\mathbf{Z}_i}, \boldsymbol{s}'_i)$

Oracle $\mathsf{H}(\boldsymbol{a})$:
$\boldsymbol{R}'_{\boldsymbol{a}} \leftarrow \sum_{j=1}^{Q_{\mathsf{P}_2}} \boldsymbol{a}_j \boldsymbol{R}_j$
$(\boldsymbol{vSid}_{\boldsymbol{a}}, \boldsymbol{c}'_{\boldsymbol{a}}) \xleftarrow{\$} \mathtt{V}_1(\boldsymbol{R}'_{\boldsymbol{a}})$
Return $\boldsymbol{c}'_{\boldsymbol{a}}$

Fig. 5. Adversary B in the $\mathbf{OMMIM}_{\mathsf{ID}}^{\mathsf{B}}$ game

in this manner, each query to H prompts B to open a session with the verifier in $\mathbf{OMMIM}_{\mathsf{ID}}$. Finally, from A's solution to the ROS problem, B successfully closes $Q_{\mathsf{P}_2}+1$ (out of Q) sessions with the verifier.

If A is successful then $\boldsymbol{c}_{Q_{\mathsf{P}_2}+1} = -1$ and $\wedge$ $\mathbf{A}\boldsymbol{c} = 0$. Furthermore for all $i \in [Q_{\mathsf{P}_2}+1]$, $\mathsf{H}(\mathbf{Z}_i) = \mathbf{A}_{i,Q_{\mathsf{P}_2}+1}$ and we have

$$\begin{aligned}\mathsf{F}(\boldsymbol{s}'_i) =& \mathsf{F}(\sum_{j=1}^{Q_{\mathsf{P}_2}} \mathbf{A}_{i,j}\boldsymbol{s}_j) = \sum_{j=1}^{Q_{\mathsf{P}_2}} \mathbf{A}_{i,j}(\boldsymbol{c}_j \cdot pk + \boldsymbol{R}_j) = pk \sum_{j=1}^{Q_{\mathsf{P}_2}} \mathbf{A}_{i,j}\boldsymbol{c}_j + \boldsymbol{R}'_{\mathbf{Z}_i} \\ =& pk \cdot \boldsymbol{c}'_{\mathbf{Z}_i} + \boldsymbol{R}'_{\mathbf{Z}_i},\end{aligned}$$

which is equivalent to $\mathsf{Ver}(pk, \boldsymbol{R}'_{\mathbf{Z}_i}, \boldsymbol{c}'_{\mathbf{Z}_i}, \boldsymbol{s}'_i) = 1$. This shows $\boldsymbol{b}_i = 1$ for all $i \in [Q_{\mathsf{P}_2}+1]$, which concludes the proof.

5 Canonical Blind Signature Schemes

5.1 Syntax of Canonical Blind Signature Schemes

We now introduce the syntax of a canonical blind signature scheme. We use the term canonical to describe a three-move blind signature protocol in which the signer's and the user's moves consist of picking and sending a random strings of some length, and the user's final signature is a deterministic function of the conversation and the public key. For simplicity, we assume the existence of a public set of parameters *par*.

Definition 2 (Canonical Blind Signature Scheme). *A* canonical blind signature scheme BS *is a tuple of algorithms* $\mathsf{BS} = (\mathsf{KG}, \mathsf{S}, \mathsf{U}, \mathsf{Ver})$.

– *The* key generation algorithm KG *outputs a public key/secret key pair* (pk, sk)*. We assume that pk implicitly defines a challenge set* $\mathcal{C} = \mathcal{C}(pk)$*.*

- *The Signer algorithm* S *is split into two algorithms* $\mathsf{S} = (\mathsf{S}_1, \mathsf{S}_2)$*.* S_1 *returns the first message of the transcript, commitment* R *and the Signers's state* st_{S}*. Deterministic algorithm* S_2 *takes as input the Signer's state* st_{S}*, a secret key* sk*, a commitment* R*, and a challenge* $c \in \mathcal{C}$*. It returns with the last message of the transcript, the answer* s*.*
- *The User algorithm* U *is split into two algorithms* $\mathsf{U} = (\mathsf{U}_1, \mathsf{U}_2)$*.* U_1 *takes as input the public key* pk*, a commitment* R*, a message* m *and returns the Users' state* st_{U} *and the second message of the transcript, a challenge* $c \in \mathcal{C}$*. Deterministic algorithm* U_2 *takes as input the public key* pk*, the transcript* (R, c, s)*, a message* m*, the Users' state* st_{U} *and outputs a signature* σ*.*
- *The deterministic verification algorithm* Ver *takes as input a message* m*, a signature* σ*, a public key* pk *and outputs a bit* b *indicating accept* $(b = 1)$ *or reject* $(b = 0)$*.*

The diagram below depicts an interaction between signer S *and user* U*. For perfect correctness we require that for all* $(pk, sk) \xleftarrow{\$} \mathsf{KG}(par)$*,* $m \in \{0,1\}^*$*,* σ *being the output of the interaction of* $\mathsf{S}(sk)$ *and* $\mathsf{U}(pk, m)$ *we have* $\mathsf{Ver}(pk, \sigma, m) = 1$*.*

Signer $\mathsf{S}(sk)$		User $\mathsf{U}(pk, m)$
$(st_{\mathsf{S}}, R) \xleftarrow{\$} \mathsf{S}_1(sk)$	$\xrightarrow{R}$	
	$\xleftarrow{c}$	$(st_{\mathsf{U}}, c) \xleftarrow{\$} \mathsf{U}_1(pk, R, m)$
$s \leftarrow \mathsf{S}_2(sk, R, c, st_{\mathsf{S}})$	$\xrightarrow{s}$	$\sigma \leftarrow \mathsf{U}_2(pk, R, c, s, m, st_{\mathsf{U}})$
Output 1		Output σ

We remark that modeling S_2 and U_2 as deterministic algorithms is w.l.o.g. since randomness can be transmitted through the states.

5.2 Security of canonical blind signature schemes

Security of a Canonical Blind Signature Scheme BS is captured by two security notions: *blindness* and *one more unforgability.*

Blindness. Intuitively, blindness ensures that a signer S that issues signatures on two messages $(\boldsymbol{m}_0, \boldsymbol{m}_1)$ of its own choice to a user U, can not tell in what order it issues them. In particular, S is given both resulting signatures $\boldsymbol{\sigma}_0, \boldsymbol{\sigma}_1$, and gets to keep the transcripts of both interactions with U. Let A be an adversary in the $\mathbf{Blind}_{\mathsf{BS}}^{\mathsf{A}}$ experiment. In BS, the experiment takes the role of an User and A takes the role of the signer. First, the experiment selects a random bit b which will decide the order of adversarially chosen messages in both transcripts. Then A is given access to all three oracles $\mathtt{Init}, \mathtt{U}_1$ and $\mathtt{U}_2$. By convention, A first has to query oracle $\mathtt{Init}$. Then, by the definition of the experiment, A may query at most two sessions. During these two sessions A learns two sets of transcripts $\boldsymbol{T}_0 = (\boldsymbol{R}_0, \boldsymbol{c}_0, \boldsymbol{s}_0)$ and $\boldsymbol{T}_1 = (\boldsymbol{R}_1, \boldsymbol{c}_1, \boldsymbol{s}_1)$. In transcripts $\boldsymbol{T}_0$ and $\boldsymbol{T}_1$, the experiment embeds messages $\boldsymbol{m}_b$ and $\boldsymbol{m}_{1-b}$, respectively. If A behaves honestly, A learns signatures $\boldsymbol{\sigma}_b$ and $\boldsymbol{\sigma}_{1-b}$ on messages $\boldsymbol{m}_b$ and $\boldsymbol{m}_{1-b}$, else nothing at all. At the end of the experiment, for A to win, A has to guess

GAME $\mathbf{Blind}^{\mathsf{A}}_{\mathsf{BS}}$:

$b \xleftarrow{\$} \{0,1\}$; $\boldsymbol{b}_1 \leftarrow b$; $\boldsymbol{b}_2 \leftarrow 1-b$

$b' \xleftarrow{\$} \mathsf{A}^{\mathtt{Init},\mathtt{U}_1,\mathtt{U}_2}()$

Return $b = b'$

Oracle $\mathtt{Init}(pk, \boldsymbol{m}_0, \boldsymbol{m}_1)$ // one, first query

Absorb pk as public key

$\mathbf{sess}_1 \leftarrow \mathbf{sess}_2 \leftarrow \mathtt{init}$

Oracle $\mathtt{U}_1(sid, R)$

If $sid \notin \{1,2\} \vee \mathbf{sess}_{sid} \neq \mathtt{init}$ Then

  Return $\perp$ // max. two sessions

$\mathbf{sess}_{sid} \leftarrow \mathtt{open}$

$\boldsymbol{R}_{sid} \leftarrow R$

$(\boldsymbol{st}_{sid}, \boldsymbol{c}_{sid}) \xleftarrow{\$} \mathsf{U}_1(pk, \boldsymbol{R}_{sid}, \boldsymbol{m}_{\boldsymbol{b}_{sid}})$

Return $(sid, \boldsymbol{c}_{sid})$

Oracle $\mathtt{U}_2(sid, s)$

If $\mathbf{sess}_{sid} \neq \mathtt{open}$ Then

  Return $\perp$

$\mathbf{sess}_{sid} \leftarrow \mathtt{closed}$

$\boldsymbol{s}_{sid} \leftarrow s$

$\boldsymbol{\sigma}_{\boldsymbol{b}_{sid}} \xleftarrow{\$} \mathsf{U}_2(pk, \boldsymbol{st}_{sid}, \boldsymbol{R}_{sid}, \boldsymbol{c}_{sid}, \boldsymbol{s}_{sid})$

If $\mathbf{sess}_1 = \mathbf{sess}_2 = \mathtt{closed}$ Then

  If $\boldsymbol{\sigma}_0 = \perp \vee \boldsymbol{\sigma}_1 = \perp$ Then

    $(\boldsymbol{\sigma}_0, \boldsymbol{\sigma}_1) := (\perp, \perp)$

  Return $(\boldsymbol{\sigma}_0, \boldsymbol{\sigma}_1)$ // return both signatures

Else

  Return ε

Fig. 6. Games defining $\mathbf{Blind}^{\mathsf{A}}_{\mathsf{BS}}$ for a canonical blind signature scheme BS, with the convention that A makes exactly one query to $\mathtt{Init}$ at the beginning of its execution.

the bit b. In Figure 6 we formally define the $\mathbf{Blind}^{\mathsf{A}}_{\mathsf{BS}}$ experiment. Formally, the advantage function of an adversary A in attacking the blindness of BS is defined as $\mathbf{Adv}^{\mathbf{Blind}}_{\mathsf{BS}}(\mathsf{A}) := \Pr[\mathbf{Blind}^{\mathsf{A}}_{\mathsf{BS}} \Rightarrow 1] - \frac{1}{2}$. We say BS is *perfectly blind* if $\mathbf{Adv}^{\mathbf{Blind}}_{\mathsf{BS}}(\mathsf{A}) = 0$.

OMUF-Security of Blind Signature Schemes. We now define the standard unforgeability notion for blind signatures, namely *one-more unforgeability*. Intuitively, One-More Unforgeability ensures that a user U can not produce a single signature more than it should be able to learn from interactions with the signer S. Let A be an adversary in the $\mathbf{OMUF}^{\mathsf{A}}_{\mathsf{BS}}$ experiment, which takes the role of the User. Let $Q_{\mathsf{S}} \leftarrow Q_{\mathsf{S}_1} + Q_{\mathsf{S}_2}$. Session identifier $sid \in [Q_{\mathsf{S}}]$ is used to model one interaction with the signer. A call to $\mathtt{S}_1$ returns a new session identifier $sid \in [Q_{\mathsf{S}}]$ and sets flag $\mathbf{sess}_{sid}$ to $\mathtt{open}$. A call to $\mathtt{S}_2(sid, \cdot)$ with the same sid sets the flag $\mathbf{sess}_{sid}$ to $\mathtt{closed}$. The closed sessions result in Q_{S_2} different transcripts $(\boldsymbol{R}_k, \boldsymbol{c}_k, \boldsymbol{s}_k)$, where each challenge $\boldsymbol{c}_i$ is adversarially chosen. (The remaining Q_{S_1} abandoned sessions are of the form $(\boldsymbol{R}_k, \perp, \perp)$ and hence do not contain a complete transcript.) A wins the experiment, if it is able to produce $\ell(\mathsf{A}) \geq Q_{\mathsf{S}_2}(\mathsf{A}) + 1$ signatures (on distinct messages) after having interacted with $Q_{\mathsf{S}_2}(\mathsf{A}) \leq Q_{\mathsf{S}_2}$ closed signer sessions (from which he should be able to compute ℓ signatures). In Figure 7 we formally define the $\mathbf{OMUF}^{\mathsf{A}}_{\mathsf{BS}}$ experiment. Formally, the advantage function of an adversary A in attacking the One-More Unforgeability of BS is defined as $\mathbf{Adv}^{\mathbf{OMUF}}_{\mathsf{BS}}(\mathsf{A}) := \Pr[\mathbf{OMUF}^{\mathsf{A}}_{\mathsf{BS}} \Rightarrow 1]$.

We say that BS is $(\varepsilon, t, Q_{\mathsf{S}_1}, Q_{\mathsf{S}_2})$-**OMUF** secure if for all adversaries A satisfying $\mathbf{Time}(\mathsf{A}) \leq t$, $Q_{\mathsf{S}_2}(\mathsf{A}) \leq Q_{\mathsf{S}_2}$, and $Q_{\mathsf{S}_1}(\mathsf{A}) \leq Q_{\mathsf{S}_1}$, we have $\mathbf{Adv}^{\mathbf{OMUF}}_{\mathsf{BS}}(\mathsf{A}) \leq \varepsilon$. In the random oracle model we say BS is $(\varepsilon, t, Q_{\mathsf{S}_1}, Q_{\mathsf{S}_2}, Q_{\mathsf{H}})$-**OMUF** secure if for all adversaries A variables $\varepsilon, t, Q_{\mathsf{S}_1}$ and Q_{S_2} satisfy the latter conditions and Q_{H} is the number of queries to H.

GAME $\mathbf{OMUF}_{\mathsf{BS}}^{\mathsf{A}}$:

$(sk, pk) \leftarrow \mathsf{KG}(par)$
$sid \leftarrow 0$ // initialize signer session id
$((\boldsymbol{m}_1, \boldsymbol{\sigma}_1), ..., (\boldsymbol{m}_{\ell(\mathsf{A})}, \boldsymbol{\sigma}_{\ell(\mathsf{A})})) \leftarrow \mathsf{A}^{\mathsf{S}_1, \mathsf{S}_2}(pk)$
If $\exists i \neq j : \boldsymbol{m}_i = \boldsymbol{m}_j$ Then // all messages have to be distinct
  Return 0
If $\exists i \in [\ell(\mathsf{A})] : \mathsf{Ver}(pk, \boldsymbol{m}_i, \boldsymbol{\sigma}_i) = 0$ Then // all signatures have to be valid
  Return 0
$Q_{\mathsf{S}_1}(\mathsf{A}) \leftarrow \#\{k \mid \mathbf{sess}_k = \texttt{open}\}$ // #abandoned signer sessions
$Q_{\mathsf{S}_2}(\mathsf{A}) \leftarrow \#\{k \mid \mathbf{sess}_k = \texttt{closed}\}$ // #closed signer sessions
If $\ell(\mathsf{A}) \geq Q_{\mathsf{S}_2}(\mathsf{A}) + 1$ Then
  Return 1
Return 0

Oracle S_1
$sid \leftarrow sid + 1$
$\mathbf{sess}_{sid} \leftarrow \texttt{open}$
$(\boldsymbol{st}_{sid}, \boldsymbol{R}_{sid}) \xleftarrow{\$} \mathsf{S}_1(sk)$
Return $(sid, \boldsymbol{R}_{sid})$

Oracle $\mathsf{S}_2(sid, c)$
If $\mathbf{sess}_{sid} \neq \texttt{open}$ Then
  Return $\perp$
$\mathbf{sess}_{sid} = \texttt{closed}$
$\boldsymbol{s}_{sid} \leftarrow \mathsf{S}_2(sk, \boldsymbol{st}_{sid}, \boldsymbol{R}_{sid}, c)$
Return $\boldsymbol{s}_{sid}$

Fig. 7. $\mathbf{OMUF}_{\mathsf{BS}}^{\mathsf{A}}$ Game

5.3 Linear Blind Signature Schemes

Let LF be a linear function family and H a random oracle. Figure 8 shows how to construct a blind signature scheme $\mathsf{BS}[\mathsf{LF}, \mathsf{H}]$.

Algorithm $\mathsf{KG}(par)$
$sk \xleftarrow{\$} \mathcal{D}$
$pk \leftarrow \mathsf{F}(sk)$
Return (sk, pk)

Algorithm $\mathsf{S}_1(sk)$
$r \xleftarrow{\$} \mathcal{D}$; $R \leftarrow \mathsf{F}(r)$
$st_{\mathsf{S}} := r$
Return (st_{S}, R)

Algorithm $\mathsf{S}_2(sk, st_{\mathsf{S}}, c)$
$r \leftarrow st_{\mathsf{S}}$
$s \leftarrow c \cdot sk + r$
Return s

Algorithm $\mathsf{U}_1(pk, R, m)$
$\alpha \xleftarrow{\$} \mathcal{D}, \beta \xleftarrow{\$} \mathcal{S}$
$R' \leftarrow R + \mathsf{F}(\alpha) + \beta \cdot pk$
$c' \leftarrow \mathsf{H}(R', m)$
$c \leftarrow c' + \beta$
$st_{\mathsf{U}} \leftarrow (\alpha, \beta)$
Return (c, st_{U})

Algorithm $\mathsf{U}_2(pk, R, c, s, m, st_{\mathsf{U}})$
$(\alpha, \beta) \leftarrow st_{\mathsf{U}}$
$R' \leftarrow R + \mathsf{F}(\alpha) + \beta \cdot pk$
$c' \leftarrow \mathsf{H}(R', m)$
$s' \leftarrow s + \alpha$
Return $\sigma \leftarrow (c', s')$

Algorithm $\mathsf{Ver}(pk, m, \sigma)$
$(c', s') \leftarrow \sigma$
$R' \leftarrow \mathsf{F}(s') - c' \cdot pk$
If $c' \neq \mathsf{H}(R', m)$ Then
  Return 0
Return 1

Fig. 8. Let LF be a linear function and $\mathsf{H} : \{0,1\}^* \to \mathcal{C}$ be a hash function. This figure shows the construction of the canonical blind signature scheme $\mathsf{BS}[\mathsf{LF}, \mathsf{H}] = (\mathsf{KG}, \mathsf{S} = (\mathsf{S}_1, \mathsf{S}_2), \mathsf{U} = (\mathsf{U}_1, \mathsf{U}_2), \mathsf{Ver})$.

Theorem 3. *Let* LF *be a linear function family and* H *be a random oracle.* $\mathsf{ID}[\mathsf{LF}]$ *is* $(\varepsilon', t', Q_{\mathsf{V}}, Q_{\mathsf{P}_1}, Q_{\mathsf{P}_2})$-**OMMIM** *secure if and only if* $\mathsf{BS}[\mathsf{LF}, \mathsf{H}]$ *is* $(\varepsilon, t, Q_{\mathsf{S}_1}, Q_{\mathsf{S}_2}, Q_{\mathsf{H}})$-**OMUF** *secure , where*

$$t' = t, \quad \varepsilon' = \varepsilon, \quad Q_{\mathsf{V}} = Q_{\mathsf{H}} + Q_{\mathsf{S}_2} + 1, \quad Q_{\mathsf{P}_1} = Q_{\mathsf{S}_1}, \quad Q_{\mathsf{P}_2} = Q_{\mathsf{S}_2} \ .$$

Proof. Let A be an $(\varepsilon, t, Q_{\mathsf{S}_1}, Q_{\mathsf{S}_2}, Q_{\mathsf{H}})$-**OMUF** adversary in the $\mathbf{OMUF}_{\mathsf{BS}}$ experiment. In Figure 9 we construct an $(\varepsilon', t', Q_{\mathsf{V}}, Q_{\mathsf{P}_1}, Q_{\mathsf{P}_2})$-**OMMIM** adversary B that is executed in the $\mathbf{OMMIM}_{\mathsf{ID}}$ experiment that perfectly simulates A's oracles $\mathtt{S}_1$, $\mathtt{S}_2$ and H via its own oracles $\mathtt{P}_1$, $\mathtt{P}_2$, and $\mathtt{V}_1$, respectively. Suppose that A is successful, i.e., it outputs $Q_{\mathsf{P}_2} + 1$ valid signatures on distinct messages and the number of successfully sessions with the signer is at most Q_{P_2}. Since $\boldsymbol{\sigma}_i$ is a valid signature on $\boldsymbol{m}_i$, B can make a successful query to oracle $\mathtt{V}_2(vSid, \boldsymbol{s}'_i)$ in line 06 resulting in $\boldsymbol{b}_i = 1$. Overall, B makes $Q_{\mathsf{P}_2} + 1$ successful queries to $\mathtt{V}_2$ such that the internal counter $\ell(\mathsf{A})$ is set to $Q_{\mathsf{P}_2} + 1$ and B wins. This proves $\varepsilon' \geq \varepsilon$. Moreover, the number of abandoned sessions (denoted as Q_{S_1}) in the $\mathbf{OMUF}_{\mathsf{BS}}$ experiment equals the number of abandoned sessions (denoted as Q_{P_1}) in the $\mathbf{OMMIM}_{\mathsf{ID}}$ experiment and the number of calls to oracle $\mathtt{V}_1$ is bounded by Q_{H} plus additional $Q_{\mathsf{P}} + 1$ implicit calls in Line 04.

Adversary $\mathsf{B}^{\mathtt{P}_1, \mathtt{P}_2, \mathtt{V}_1, \mathtt{V}_2}(pk)$:
00 $((\boldsymbol{m}_1, \boldsymbol{\sigma}_1), \ldots, (\boldsymbol{m}_{Q_{\mathsf{P}_2}+1}, \boldsymbol{\sigma}_{Q_{\mathsf{P}_2}+1})) \leftarrow \mathsf{A}^{\mathtt{S}_1, \mathtt{S}_2, \mathsf{H}}(pk)$
01 For $i \in [Q_{\mathsf{P}_2} + 1]$ do
02 $\quad (\boldsymbol{c}'_i, \boldsymbol{s}'_i) \leftarrow \boldsymbol{\sigma}_i$
03 $\quad \boldsymbol{R}'_i \leftarrow \mathsf{F}(\boldsymbol{s}'_i) - \boldsymbol{c}'_i \cdot pk$
04 $\quad \boldsymbol{h}_i \leftarrow \mathsf{H}(\boldsymbol{R}'_i, \boldsymbol{m}_i)$
05 $\quad vSid \leftarrow \boldsymbol{SID}_{(\boldsymbol{R}'_i, \boldsymbol{m}_i)}$
06 $\quad \boldsymbol{b}_i \leftarrow \mathtt{V}_2(vSid, \boldsymbol{s}'_i)$

Oracle $\mathtt{S}_1$
07 $(pSid, \boldsymbol{R}_{pSid}) \xleftarrow{\$} \mathtt{P}_1$
08 Return $(pSid, \boldsymbol{R}_{pSid})$

Oracle $\mathtt{S}_2(pSid, c)$
09 $\boldsymbol{s}_{pSid} \leftarrow \mathtt{P}_2(pSid, c)$
10 Return $\boldsymbol{s}_{pSid}$

Oracle $\mathsf{H}(R', m)$
11 if $\boldsymbol{H}_{(R',m)} \neq \perp$ Then
12 $\quad$ Return $\boldsymbol{H}_{(R',m)}$
13 $(vSid, h) \xleftarrow{\$} \mathtt{V}_1(R')$
14 $\boldsymbol{SID}_{(R',m)} \leftarrow vSid$
15 Return $\boldsymbol{H}_{(R',m)} \leftarrow h$

Fig. 9. Reduction from $\mathbf{OMMIM}^{\mathsf{B}}_{\mathsf{ID}}$ to $\mathbf{OMUF}^{\mathsf{A}}_{\mathsf{BS}}$

Let B be an $(\varepsilon, t, Q_{\mathsf{V}}, Q_{\mathsf{P}_1}, Q_{\mathsf{P}_2})$-**OMMIM** adversary in the $\mathbf{OMMIM}_{\mathsf{ID}}$ experiment. In Figure 10 we construct an $(\varepsilon', t', Q_{\mathsf{S}_1}, Q_{\mathsf{S}_2}, Q_{\mathsf{H}})$-**OMUF** adversary A that is executed in the $\mathbf{OMUF}_{\mathsf{BS}}$ experiment that perfectly simulates B's oracles $\mathtt{P}_1, \mathtt{P}_2$ and $\mathtt{V}_1$ via its own oracles $\mathtt{S}_1, \mathtt{S}_2$ and H, respectively. To simulate oracle $\mathtt{V}_2$, A executes the same code as specified in the $\mathbf{OMMIM}_{\mathsf{ID}}$ experiment, with the only difference being line 20. This additional line does not change the behavior of $\mathtt{V}_2$ and is thus not detectable by B. Suppose that B is successful, i.e., it completes Q_{P_2} sessions as a verifier and $Q_{\mathsf{P}_2} + 1$ sessions as a prover (denoted as $\ell(\mathsf{B})$ in the $\mathbf{OMMIM}_{\mathsf{ID}}$ experiment). From the $Q_{\mathsf{P}_2} + 1$ successful calls of B to $\mathtt{V}_2$, it follows that A learns $Q_{\mathsf{P}_2} + 1$ transcripts $(\boldsymbol{R}, \boldsymbol{c}, \boldsymbol{s})$ from the view of an honest User in BS. Since messages $\boldsymbol{m}$ are constructed by calling U_1, A creates $Q_{\mathsf{P}_2} + 1$ signatures after learning values $\boldsymbol{s}$ by simply following the protocol specification of

U_2. This proves $\varepsilon' \geq \varepsilon$. Moreover the number of abandoned sessions (denoted as $Q_{\mathsf{P}_1}(\mathsf{B})$) in the $\mathbf{OMMIM}_{\mathsf{ID}}$ experiment equals the number of abandoned sessions (denoted as $Q_{\mathsf{S}_1}(\mathsf{A})$) in the $\mathbf{OMUF}_{\mathsf{BS}}$ experiment.

Adversary $\mathsf{A}^{\mathtt{S}_1,\mathtt{S}_2,\mathsf{H}}(pk)$:
$vSid \leftarrow 0$
$\mathsf{B}^{\mathtt{P}_1,\mathtt{P}_2,\mathtt{V}_1,\mathtt{V}_2}(pk)$
$i \leftarrow 1$
For all k where $\mathbf{vSess}_k = \mathtt{closed}$:
  $\boldsymbol{m}_i \leftarrow k, \boldsymbol{\sigma}_i \leftarrow (\boldsymbol{c}'_k := \boldsymbol{c}_k - \boldsymbol{\beta}_k, \boldsymbol{s}'_k := \boldsymbol{s}_k + \boldsymbol{\alpha}_k)$
  $i \leftarrow i + 1$
Return $(\boldsymbol{m}_1, \boldsymbol{\sigma}_1), \ldots, (\boldsymbol{m}_{\ell+1}, \boldsymbol{\sigma}_{\ell+1})$

Oracle $\mathtt{P}_1$
$(pSid, \boldsymbol{R}_{pSid}) \xleftarrow{\$} \mathtt{S}_1$
Return $(pSid, \boldsymbol{R}_{pSid})$

Oracle $\mathtt{P}_2(pSid, c)$
$\boldsymbol{s}_{pSid} \leftarrow \mathtt{S}_2(pSid, c)$
Return $\boldsymbol{s}_{pSid}$

Oracle $\mathtt{V}_1(R)$
$vSid \leftarrow vSid + 1$
$\mathbf{vSess}_{vSid} \leftarrow \mathtt{open}$
$(\boldsymbol{c}_{vSid}, \boldsymbol{st}_{vSid}) \leftarrow \mathsf{U}_1(pk, R, m := vSid)$
$(\boldsymbol{\alpha}_{vSid}, \boldsymbol{\beta}_{vSid}) \leftarrow \boldsymbol{st}_{vSid}$
Return $(vSid, \boldsymbol{c}_{vSid})$

Oracle $\mathtt{V}_2(vSid, s)$
If $\mathbf{vSess}_{vSid} \neq \mathtt{open}$ Then
  Return $\perp$
$\boldsymbol{b}_{vSid} \leftarrow \mathsf{Ver}(pk, \boldsymbol{R}_{vSid}, \boldsymbol{c}_{vSid}, s)$
$\mathbf{vSess}_{vSid} \leftarrow \mathtt{closed}$
$\boldsymbol{s}_{vSid} \leftarrow s$
Return $\boldsymbol{b}_{vSid}$

Fig. 10. Reduction from $\mathbf{OMUF}^{\mathsf{A}}_{\mathsf{BS}}$ to $\mathbf{OMMIM}^{\mathsf{B}}_{\mathsf{ID}}$

Theorem 4. *If* LF *is a linear function, then* $\mathsf{BS}[\mathsf{LF}, \mathsf{H}]$ *is perfectly blind.*

Proof. Let A be an adversary playing in game $\mathbf{Blind}^{\mathsf{A}}_{\mathsf{BS}[\mathsf{LF},\mathsf{H}]}$. After its execution, A holds $(\boldsymbol{m}_0, \boldsymbol{\sigma}_0)$, $(\boldsymbol{m}_1, \boldsymbol{\sigma}_1)$ where $\boldsymbol{\sigma}_0$ is a signature on $\boldsymbol{m}_0$ and $\boldsymbol{\sigma}_1$ is a signature on $\boldsymbol{m}_1$. (Here we assume without loss of generality that both signatures are valid as otherwise A obtains $\boldsymbol{\sigma}_0 = \boldsymbol{\sigma}_1 = \perp$ and thus $\mathbf{Adv}^{\mathsf{A}}_{\mathbf{Blind},\mathsf{BS}[\mathsf{LF},\mathsf{H}]} = 0$.) Adversary A furthermore learns two transcripts $\boldsymbol{T}_1 = (\boldsymbol{R}_1, \boldsymbol{c}_1, \boldsymbol{s}_1)$ and $\boldsymbol{T}_2 = (\boldsymbol{R}_2, \boldsymbol{c}_2, \boldsymbol{s}_2)$ from its interaction with the first and the second signer session, respectively. The goal of A is to match the message/signature pairs with the two transcripts.

We show that there exists no adversary which is able to distinguish, whether the message $\boldsymbol{m}_0$ was used by the experiment to create Transcript $\boldsymbol{T}_1$ or $\boldsymbol{T}_2$. We argue that for all sessions $1 \leq i \leq 2$ and indexes $0 \leq j \leq 1$, the tuple $(\boldsymbol{T}_i, \boldsymbol{m}_j, \boldsymbol{\sigma}_j)$ completely determines $\boldsymbol{st}_j = (\boldsymbol{\alpha}_{(i,j)}, \boldsymbol{\beta}_{(i,j)})$. This implies that given A's view, it is equally likely that the experiment was executed with $b = 0$ or $b = 1$ since for both choices $b \in \{0, 1\}$ there exists properly distributed states $(\boldsymbol{st}_0, \boldsymbol{st}_1)$ that would have resulted in A's view.

It remains to argue that $\boldsymbol{T}_i = (\boldsymbol{R}_i, \boldsymbol{c}_i, \boldsymbol{s}_i)$, $\boldsymbol{m}_j$, and $\boldsymbol{\sigma}_j = (\boldsymbol{c}'_j, \boldsymbol{s}'_j)$ determine values $\boldsymbol{\alpha}_{(i,j)}, \boldsymbol{\beta}_{(i,j)}$ such that $\boldsymbol{c}'_j = \mathsf{H}(\boldsymbol{R}_i + \boldsymbol{\beta}_{(i,j)} \cdot pk + \mathsf{F}(\boldsymbol{\alpha}_{(i,j)}), \boldsymbol{m}_j)$ and $\boldsymbol{\alpha}_{(i,j)} = \boldsymbol{s}'_j - \boldsymbol{s}_i, \boldsymbol{\beta}_{(i,j)} = \boldsymbol{c}_i - \boldsymbol{c}'_j$. Uniformity of $(\boldsymbol{\alpha}_{(i,j)}, \boldsymbol{\beta}_{(i,j)})$ is implied by uniformity of $(\boldsymbol{s}'_j, \boldsymbol{c}'_j)$, which come from the experiment.

Since $\boldsymbol{T}_i$ is a valid transcript, we have $\mathsf{F}(\boldsymbol{s}_i) = \boldsymbol{R}_i + \boldsymbol{c}_i \cdot pk$. Therefore

$$\begin{aligned}\boldsymbol{R}_i + \boldsymbol{\beta}_{(i,j)} \cdot pk + \mathsf{F}(\boldsymbol{\alpha}_{(i,j)}) &= \boldsymbol{R}_i + (\boldsymbol{c}_i - \boldsymbol{c}'_j) \cdot pk + \mathsf{F}(\boldsymbol{s}'_j - \boldsymbol{s}_i) \\ &= \boldsymbol{R}_i + \boldsymbol{c}_i \cdot pk - \mathsf{F}(\boldsymbol{s}_i) + \mathsf{F}(\boldsymbol{s}'_j) - \boldsymbol{c}'_j \cdot pk \\ &= \mathsf{F}(\boldsymbol{s}'_j) - \boldsymbol{c}'_j \cdot pk \;.\end{aligned}$$

Since $\boldsymbol{\sigma}_j$ is a valid signature on $\boldsymbol{m}_j$ we have $\mathsf{H}(\mathsf{F}(\boldsymbol{s}'_j) - \boldsymbol{c}'_j \cdot pk, \boldsymbol{m}_j) = \boldsymbol{c}'_j$ which concludes the proof.

Corollary 1. *Let* LF *be a linear function family with a torsion-free element from the kernel. If* LF *is* (ε', t')-**CR** *secure, then* $\mathsf{BS}[\mathsf{LF}, \mathsf{H}]$ *is* $(\varepsilon, t, Q_{\mathsf{S}_1}, Q_{\mathsf{S}_2}, Q_{\mathsf{H}})$-**OMUF** *secure where*

$$t' = 2t, \quad \varepsilon' = O\left(\left(\varepsilon - \frac{(Q+Q_{\mathsf{S}})^{Q_{\mathsf{S}_2}+1}}{q}\right)\frac{1}{Q^2 Q_{\mathsf{S}_2}^3}\right),$$

$Q_{\mathsf{S}} = Q_{\mathsf{S}_2} + Q_{\mathsf{S}_1}$ *and* $Q = Q_{\mathsf{H}} + Q_{\mathsf{S}_2} + 1$. *Moreover,* $\mathsf{BS}[\mathsf{LF}, \mathsf{H}]$ *is perfectly blind.*

Proof. The proof of the one-more unforgability security follows from combining Theorems 1 and 3. Perfect blindness follows directly from Theorem 4.

6 Proof of Theorem 1

Before we give the proof of Theorem 1, we provide some intuition about the difficulty that arises in the context of proving the **OMMIM**-security of $\mathsf{ID}[\mathsf{LF}]$ and how our proof overcomes it. The main issue is that the adversary M in **OMMIM** can interleave sessions between the oracles $\mathsf{P}_1, \mathsf{P}_2$ and $\mathtt{V}_1, \mathtt{V}_2$. This gives M strong adaptive capabilities which lead to the ROS-attack described in 4.2. The ROS-attack is reflected in Corollary 2, which can be translated into an upper bound on M's success probability of providing our reduction with two identical values $\hat{\chi}, \hat{\chi}'$ that result from running the adversary twice with fixed public key pk and randomness ω, but (partially) different replies $\boldsymbol{h}, \boldsymbol{h}'$ to $\mathtt{V}_1$. If the adversary succeeds in setting $\hat{\chi} = \hat{\chi}'$, the reduction fails in recovering a collision with respect to LF, i.e., values $\hat{\chi} \neq \hat{\chi}'$ s.t. $\mathsf{LF}(\hat{\chi}) = \mathsf{LF}(\hat{\chi}')$.

To prove the bound in Corollary 2, our proof follows the ideas of [21], but takes into account also the abandoned sessions with P_1, which [21] does not consider. The intuitive idea behind ensuring $\hat{\chi} \neq \hat{\chi}'$ is to run M on an instance $I = pk$ that could be the result of applying F to either sk or $\hat{sk} = sk + z^*$ from the domain $\mathcal{D}$ of F. One can show that from M's perspective, the resulting view is identical in both cases (Lemma 7). On the other hand, since $\hat{\chi}$ depends non-trivially on sk (or $\hat{sk}$, respectively), it should take (with high probability) different values from the reduction's point of view, depending on whether the reduction used sk or $sk + z^*$

as a preimage to pk. Indeed, this intuition is supported by Corollary 2. However, Corollary 2 can only be translated into an upper bound on the probability that $\hat{\chi}$ takes the same *particular* value $C(sk, \omega, \boldsymbol{h})$, regardless of whether sk or $\hat{sk}$ was used by the reduction. Intuitively, $C(sk, \omega, \boldsymbol{h})$ is the value that is most likely taken by the random variable $\hat{\chi}'$, which occurs as the result of rewinding M with the same sk, ω, but a partially different set of $\mathtt{V}_1$-replies $\boldsymbol{h}'$ (i.e., the probability is over the fresh values in $\boldsymbol{h}'$). To ensure that $\hat{\chi} \neq \hat{\chi}'$, the analysis first defines the set $\mathcal{B}$ of tuples $(sk, \omega, \boldsymbol{h})$ which yield a successful run of M, but for which $\hat{\chi}(sk, \omega, \boldsymbol{h}) \neq C(sk, \omega, \boldsymbol{h})$. It then estimates the probability that both tuples $(sk, \omega, \boldsymbol{h}), (sk, \omega, \boldsymbol{h}')$ that are used to run M, are tuples from the set $\mathcal{B}$. The final step of the proof is to leverage this fact to obtain a lower bound on the success probability of the reduction, i.e., to ensure that $\hat{\chi} \neq \hat{\chi}'$ (Lemma 2). To argue that not only both runs of M are successful, but yield tuples in $\mathcal{B}$, we present a more general version of the forking lemma by Bellare and Neven [6].

6.1 The reduction algorithm

Let M be an $(\varepsilon, t, Q_{\mathsf{V}}, Q_{\mathsf{P}_1}, Q_{\mathsf{P}_2})$-**OMMIM** adversary that plays in game $\mathbf{OMMIM}_{\mathsf{ID[LF]}}$. Without loss of generality, we will assume throughout the proof that $Q_{\mathsf{P}_1}(\mathsf{M}) = Q_{\mathsf{P}_1}, Q_{\mathsf{P}_2}(\mathsf{M}) = Q_{\mathsf{P}_2}, Q_{\mathsf{V}}(\mathsf{M}) = Q_{\mathsf{V}}, \ell(\mathsf{M}) = Q_{\mathsf{P}_2} + 1$, as well as $Q_{\mathsf{P}_1} \geq Q_{\mathsf{P}_2}$.

For $1 \leq i \leq Q_{\mathsf{P}_2} + 1$, we define an auxiliary algorithm A_i which 'sandboxes' M and that will be used later by another adversary B to break collision resistance of LF. More concretely, A_i obtains as input an instance $I = sk$, runs M on random tape ω and uses vector $\boldsymbol{h} \in \mathcal{C}^{Q_{\mathsf{V}}}$ to answer M's Q_{V} queries to $\mathtt{V}_1$. The description of algorithm A_i is given in Figure 11. Note that A_i is deterministic for fixed randomness ω.

Analysis of A_i. To analyze A_i, we now introduce some notation. First, consider the variables $\hat{\boldsymbol{J}}_i, \hat{\boldsymbol{\chi}}_i, \hat{\boldsymbol{s}}'$, and $\hat{\boldsymbol{h}}_i$ defined on Lines 32 through 35 of Figure 11. These variables are introduced to simplify the referencing of values associated with successful calls to the verification oracle $\mathtt{V}_2(vSid, \cdot)$ over the course of the proof. Concretely, the variable

$$\hat{\boldsymbol{\chi}}_i = \hat{\boldsymbol{s}}'_i - \hat{\boldsymbol{h}}_i \cdot sk$$

results from the *i-th successful call* to the verification oracle $\mathtt{V}_2(vSid, \cdot)$, whereas the index $\hat{\boldsymbol{J}}_i$ indicates which session identity $vSid$ corresponds to this call.

We will fix an execution of A_i via the tuples $I = sk$, $\boldsymbol{h}$, and A_i's randomness ω. We define the set $\mathcal{W}$ of *successful inputs of* A_i as the set of all such tuples $(I, \omega, \boldsymbol{h})$ which lead to a successful run of A_i, i.e.,

$$\mathcal{W} := \{(I, \omega, \boldsymbol{h}) \mid \hat{\boldsymbol{J}}_i \neq 0; (\hat{\boldsymbol{J}}_i, \hat{\boldsymbol{\chi}}_i) \leftarrow \mathsf{A}_i(I, \boldsymbol{h}; \omega)\}$$

Note that $\mathcal{W}$ is independent of i and, by construction of A_i,

$$\Pr_{(I,\omega,\boldsymbol{h}) \xleftarrow{\$} (\mathcal{I} \times \Omega \times \mathcal{C}^{Q_{\mathsf{V}}})}[(I, \omega, \boldsymbol{h}) \in \mathcal{W}] = \mathbf{Adv}^{\mathbf{OMMIM}}_{\mathsf{ID[LF]}}(\mathsf{M}) = \varepsilon.$$

We can view $\hat{\boldsymbol{J}}_i, \hat{\boldsymbol{\chi}}_i, \hat{\boldsymbol{s}}'$, and $\hat{\boldsymbol{h}}_i$ as random variables whose distribution is induced by the the uniform distribution on $(\mathcal{I} \times \Omega \times \mathcal{C}^{Q_{\mathsf{V}}})$. Furthermore, their outcome is uniquely determined given $(I, \omega, \boldsymbol{h}) \in \mathcal{W}$, so let us write in this case

$$\left(\hat{\boldsymbol{J}}_i(I, \omega, \boldsymbol{h}), \hat{\boldsymbol{\chi}}_i(I, \omega, \boldsymbol{h})\right) \leftarrow \mathsf{A}_i(I, \boldsymbol{h}; \omega).$$

Adversary $\mathsf{A}_i(I = sk, \boldsymbol{h}; \omega)$:
Parse $(\omega_{\mathsf{M}}, \boldsymbol{r}) \leftarrow \omega$
$\boldsymbol{R} \leftarrow \mathsf{F}(\boldsymbol{r})$
$pk \leftarrow \mathsf{F}(sk)$
$ctr \leftarrow 0; pSid \leftarrow 0; vSid \leftarrow 0$
$\mathsf{M}^{\mathtt{P}_1,\mathtt{P}_2,\mathtt{V}_1,\mathtt{V}_2}(pk)$
$\ell(\mathsf{M}) \leftarrow \#\{k \mid \mathbf{vSess}_k = \mathtt{closed} \wedge \boldsymbol{b}_k = 1\}$
$Q_{\mathsf{P}_2}(\mathsf{M}) \leftarrow \#\{k \mid \mathbf{pSess}_k = \mathtt{closed}\}$
$Q_{\mathsf{P}_1}(\mathsf{M}) \leftarrow \#\{k \mid \mathbf{pSess}_k = \mathtt{open}\}$
$Q_{\mathsf{V}}(\mathsf{M}) \leftarrow vSid$
If $(\ell(\mathsf{M}) \geq Q_{\mathsf{P}_2}(\mathsf{M}) + 1)$ Then
  Return $(\hat{\boldsymbol{J}}_i, \hat{\boldsymbol{\chi}}_i)$
Return $(\hat{\boldsymbol{J}}_i, \hat{\boldsymbol{\chi}}_i) \leftarrow (0, 0)$

Procedure $\mathtt{P}_1$
$pSid \leftarrow pSid + 1$
$\mathbf{pSess}_{pSid} \leftarrow \mathtt{open}$
$\boldsymbol{c}_{pSid} \leftarrow \perp$
Return $(pSid, \boldsymbol{R}_{pSid})$

Procedure $\mathtt{P}_2(pSid, c)$
If $\mathbf{pSess}_{pSid} \neq \mathtt{open}$ Then
  Return $\perp$
$\mathbf{pSess}_{pSid} \leftarrow \mathtt{closed}$
$\boldsymbol{s}_{pSid} \leftarrow c \cdot sk + \boldsymbol{r}_{pSid}$
$\boldsymbol{c}_{pSid} \leftarrow c$
Return $\boldsymbol{s}_{pSid}$

Procedure $\mathtt{V}_1(R')$
$vSid \leftarrow vSid + 1$
$\boldsymbol{R}'_{vSid} \leftarrow R'$
$\mathbf{vSess}_{pSid} \leftarrow \mathtt{open}$
Return $(vSid, \boldsymbol{h}_{vSid})$

Procedure $\mathtt{V}_2(vSid, s')$
If $\mathbf{vSess}_{vSid} \neq \mathtt{open}$ Then
  Return $\perp$
$\boldsymbol{S}'_{vSid} \leftarrow \mathsf{F}(s')$
$\mathbf{vSess}_{vSid} \leftarrow \mathtt{closed}$
If $\boldsymbol{S}'_{vSid} = \boldsymbol{h}_{vSid} \cdot pk + \boldsymbol{R}'_{vSid}$ Then
  $ctr \leftarrow ctr + 1$
  $\hat{\boldsymbol{s}}'_{ctr} \leftarrow s'$
  $\hat{\boldsymbol{h}}_{ctr} \leftarrow \boldsymbol{h}_{vSid}$
  $\hat{\boldsymbol{\chi}}_{ctr} \leftarrow \hat{\boldsymbol{s}}'_{ctr} - \hat{\boldsymbol{h}}_{ctr} \cdot sk$
  $\hat{\boldsymbol{J}}_{ctr} \leftarrow vSid$
  $\boldsymbol{b}'_{vSid} \leftarrow 1$
Else
  $\boldsymbol{b}'_{vSid} \leftarrow 0$
Return $\boldsymbol{b}'_{vSid}$

Fig. 11. Wrapping adversaries A_i for $1 \leq i \leq Q_{\mathsf{P}_2} + 1$

In the following, when stating probability distributions over I, ω, and $\boldsymbol{h}$, unless specified differently, we will always refer to the uniform distributions. That is, $(I, \omega, \boldsymbol{h}) \xleftarrow{\$} (\mathcal{I} \times \Omega \times \mathcal{C}^{Q_{\mathsf{V}}})$.

We consider the following probability for fixed $(I, \omega, \boldsymbol{h}), j, c$ and i:

$$\Pr_{\boldsymbol{h'} \xleftarrow{\$} \mathcal{C}^{Q_{\mathsf{V}}} | \boldsymbol{h}_{[j-1]}} [\hat{\boldsymbol{J}}_i(I, \omega, \boldsymbol{h'}) = j \wedge \hat{\boldsymbol{\chi}}_i(I, \omega, \boldsymbol{h'}) = c], \tag{3}$$

where the conditional probability $\boldsymbol{h'} \xleftarrow{\$} \mathcal{C}^{Q_{\mathsf{V}}} | \boldsymbol{h}_{[j-1]}$ was introduced in Section 2.

We denote by $c_{i,j}(I, \omega, \boldsymbol{h})$ the lexicographically first value c s.t. the probability in (3) is maximized when $(I, \omega, \boldsymbol{h}), j, i$ are fixed. We then write $C_i(I, \omega, \boldsymbol{h}) = c_{i, \hat{\boldsymbol{J}}_i(I, \omega, \boldsymbol{h})}(I, \omega, \boldsymbol{h})$. For fixed i, j, let us define $\mathcal{B}_{i,j} \subset \mathcal{W}$ as

$$\mathcal{B}_{i,j} := \{(I, \omega, \boldsymbol{h}) \in \mathcal{W} \mid \hat{\boldsymbol{J}}_i(I, \omega, \boldsymbol{h}) = j \wedge \hat{\boldsymbol{\chi}}_i(I, \omega, \boldsymbol{h}) \neq C_i(I, \omega, \boldsymbol{h})\}.$$

Adversary $\mathsf{B}(par)$:

$i^* \xleftarrow{\$} [Q_{\mathsf{P}_2}+1]$
$\boldsymbol{h} \xleftarrow{\$} \mathcal{C}^{Q_{\mathsf{V}}}$
$\omega \xleftarrow{\$} \Omega$
$sk \xleftarrow{\$} \mathcal{D}$
$(\hat{\boldsymbol{J}}_{i^*}, \hat{\boldsymbol{\chi}}_{i^*}) \leftarrow \mathsf{A}_{i^*}(I = sk, \boldsymbol{h}; \omega)$ //First execution of A_{i^*}
If $\hat{\boldsymbol{J}}_{i^*} = 0$
  Return $\perp$
$\boldsymbol{h'} \xleftarrow{\$} \mathcal{C}^{Q_{\mathsf{V}}} | \boldsymbol{h}_{[\hat{\boldsymbol{J}}_{i^*}-1]}$ //Conditionally resample $\boldsymbol{h'}$
$(\hat{\boldsymbol{J}}'_{i^*}, \hat{\boldsymbol{\chi}}'_{i^*}) \leftarrow \mathsf{A}_{i^*}(I = sk, \boldsymbol{h'}; \omega)$ //Second execution of A_{i^*}
If $(\hat{\boldsymbol{J}}'_{i^*} = \hat{\boldsymbol{J}}_{i^*}) \wedge (\hat{\boldsymbol{\chi}}_{i^*} \neq \hat{\boldsymbol{\chi}}'_{i^*})$ Then
  return $(\hat{\boldsymbol{\chi}}_{i^*}, \hat{\boldsymbol{\chi}}'_{i^*})$
Return $\perp$

Fig. 12. Adversary B against CR of LF.

and

$$\beta_{i,j} = \Pr_{(I,\omega,\boldsymbol{h}) \xleftarrow{\$} (\mathcal{I} \times \Omega \times \mathcal{C}^{Q_{\mathsf{V}}})} [(I, \omega, \boldsymbol{h}) \in \mathcal{B}_{i,j}]$$

$$\delta_{i,j} = \Pr_{(I,\omega,\boldsymbol{h}) \xleftarrow{\$} (\mathcal{I} \times \Omega \times \mathcal{C}^{Q_{\mathsf{V}}}), \boldsymbol{h'} \xleftarrow{\$} \mathcal{C}^{Q_{\mathsf{V}}} | \boldsymbol{h}_{[j-1]}} \begin{bmatrix} \hat{\boldsymbol{\chi}}_i(I, \omega, \boldsymbol{h'}) \neq \hat{\boldsymbol{\chi}}_i(I, \omega, \boldsymbol{h}) \\ \wedge \hat{\boldsymbol{J}}_i(I, \omega, \boldsymbol{h}) = \hat{\boldsymbol{J}}_i(I, \omega, \boldsymbol{h'}) = j \end{bmatrix}.$$

Lemma 2. *For all* i, j*:* $\delta_{i,j} \geq \beta_{i,j} \left(\frac{\beta_{i,j}}{8} - \frac{1}{2q} \right)$.

The proof of this lemma is postponed to Section 6.3.

Lemma 3. *There exist* $i \in [Q_{\mathsf{P}_2}+1], j \in [Q_{\mathsf{V}}]$ *such that* $\beta_{i,j} > \left(\varepsilon - \frac{Q_{\mathsf{V}}^{Q_{\mathsf{P}_2}+1} \cdot \binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{q} \right) \cdot \frac{1}{2Q_{\mathsf{V}}(Q_{\mathsf{P}_2}+1)}$.

The proof of this lemma is postponed to Section 6.4.

Adversary B against CR of LF. We are now ready to describe our (ε', t')-adversary B depicted in Figure 12, which plays in the collision resistance game of LF. B works roughly as follows. It first samples randomness $\omega \xleftarrow{\$} \Omega$, a secret key $sk \xleftarrow{\$} \mathcal{D}$, a vector $\boldsymbol{h} \xleftarrow{\$} \mathcal{C}^{Q_{\mathsf{V}}}$, and an index $i^* \xleftarrow{\$} [Q_{\mathsf{P}_2}+1]$ and runs A_{i^*} on input $(I = sk, \boldsymbol{h}; \omega)$. It samples a second random vector $\boldsymbol{h'}$ as $\boldsymbol{h'} \xleftarrow{\$} \mathcal{C}^{Q_{\mathsf{V}}} | \boldsymbol{h}_{[\hat{\boldsymbol{J}}_{i^*}-1]}$ and runs A_{i^*} a second time with the same randomness ω and the same instance I, but replacing $\boldsymbol{h}$ by $\boldsymbol{h'}$. In the case that B does not abort, note that by definition of A_{i^*},

$$\begin{aligned} \mathsf{F}(\hat{\boldsymbol{\chi}}_{i^*}) &= \mathsf{F}(\hat{\boldsymbol{s}}'_{i^*} - \hat{\boldsymbol{h}}_{i^*} \cdot sk) \\ &= \boldsymbol{S}'_{\hat{\boldsymbol{J}}_{i^*}} - \boldsymbol{h}_{\hat{\boldsymbol{J}}_{i^*}} \cdot pk = \boldsymbol{R}'_{\hat{\boldsymbol{J}}_{i^*}} \end{aligned}$$

Because A_{i^*} sees identical answers for the first $\hat{\boldsymbol{J}}_{i^*} - 1$ queries to $\mathtt{V}_1$, it behaves identically in both runs until it receives the answer to the $\hat{\boldsymbol{J}}_{i^*}$-th query to $\mathtt{V}_1$. In particular, A_{i^*} poses the same $\hat{\boldsymbol{J}}_{i^*}$-th query to $\mathtt{V}_1$ which means that $\mathsf{F}(\hat{\boldsymbol{\chi}}'_{i^*}) = \boldsymbol{R}'_{\hat{\boldsymbol{J}}_{i^*}}$ and therefore also $\mathsf{F}(\hat{\boldsymbol{\chi}}_{i^*}) = \mathsf{F}(\hat{\boldsymbol{\chi}}'_{i^*})$. We now consider

$$\begin{aligned}
\varepsilon' = \mathbf{Adv}^{\mathbf{CR}}_{\mathsf{LF}}(\mathsf{B}) &= \Pr_{par \xleftarrow{\$} \mathsf{PGen}, (\hat{\boldsymbol{\chi}}_{i^*}, \hat{\boldsymbol{\chi}}'_{i^*}) \xleftarrow{\$} \mathsf{B}(par)} [\hat{\boldsymbol{\chi}}_{i^*} \neq \hat{\boldsymbol{\chi}}'_{i^*} \wedge \mathsf{F}(\hat{\boldsymbol{\chi}}_{i^*}) = \mathsf{F}(\hat{\boldsymbol{\chi}}'_{i^*})] \\
&= \sum_{j=1}^{Q_{\mathsf{V}}} \Pr[\hat{\boldsymbol{\chi}}_{i^*} \neq \hat{\boldsymbol{\chi}}'_{i^*} \wedge \mathsf{F}(\hat{\boldsymbol{\chi}}_{i^*}) = \mathsf{F}(\hat{\boldsymbol{\chi}}'_{i^*}) \wedge \hat{\boldsymbol{J}}_{i^*} = \hat{\boldsymbol{J}}'_{i^*} = j] \\
&= \sum_{j=1}^{Q_{\mathsf{V}}} \Pr[\hat{\boldsymbol{\chi}}_{i^*} \neq \hat{\boldsymbol{\chi}}'_{i^*} \wedge \hat{\boldsymbol{J}}_{i^*} = \hat{\boldsymbol{J}}'_{i^*} = j] = \sum_{j=1}^{Q_{\mathsf{V}}} \delta_{i^*,j} \\
&\geq \frac{1}{Q_{\mathsf{P}_2}+1} \cdot \max_{i \in [Q_{\mathsf{P}_2}+1]} \sum_{j=1}^{Q_{\mathsf{V}}} \delta_{i,j} \\
&\geq \max_{i,j} \frac{\beta_{i,j}}{2(Q_{\mathsf{P}_2}+1)} \left(\frac{\beta_{i,j}}{4} - \frac{1}{q} \right),
\end{aligned}$$

where for the first inequality we used that $\sum \delta_{i^*,j} = \max_i \sum \delta_{i,j}$ with probability at least $1/(Q_{\mathsf{P}_2}+1)$ and in the last step we applied Lemma 2. By Lemma 3 we finally obtain

$$\begin{aligned}
\varepsilon' &\geq \frac{\varepsilon - \frac{Q_{\mathsf{V}}^{Q_{\mathsf{P}_2}+1} \cdot \binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{q}}{32 Q_{\mathsf{V}}^2 (Q_{\mathsf{P}_2}+1)^3} \cdot \left(\varepsilon - \frac{Q_{\mathsf{V}}^{Q_{\mathsf{P}_2}+1} \cdot \binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{q} - \frac{16 Q_{\mathsf{V}}^2 (Q_{\mathsf{P}_2}+1)^2}{q} \right) \\
&= O\left(\left(\varepsilon - \frac{(Q_{\mathsf{V}} Q_{\mathsf{P}_1})^{Q_{\mathsf{P}_2}+1}}{q} \right) \frac{1}{Q_{\mathsf{V}}^2 Q_{\mathsf{P}_2}^3} \right),
\end{aligned}$$

where the last equality holds for $Q_{\mathsf{P}_1} \geq Q_{\mathsf{P}_2}$.

6.2 A Generalized Forking Lemma

In this section we will introduce our *Subset Forking Lemma*, a generalization of the forking lemma that will be useful for proving Lemma 2.

Lemma 4 (Subset Splitting Lemma). *Let $\mathcal{B} \subset \mathcal{X} \times \mathcal{Y}$ be such that*

$$\Pr_{(x,y) \xleftarrow{\$} \mathcal{X} \times \mathcal{Y}} [(x,y) \in \mathcal{B}] \geq \varepsilon.$$

For any $\alpha \leq \varepsilon$, define

$$\mathcal{B}_\alpha = \left\{ (x,y) \in \mathcal{X} \times \mathcal{Y} \mid \Pr_{y' \xleftarrow{\$} \mathcal{Y}} [(x,y') \in \mathcal{B}] \geq \varepsilon - \alpha \right\}.$$

Then

$$\Pr_{y,y' \xleftarrow{\$} \mathcal{Y}, x \xleftarrow{\$} \mathcal{X}} [(x,y') \in \mathcal{B} \wedge (x,y) \in \mathcal{B}] \geq (\varepsilon - \alpha) \cdot \alpha.$$

Proof. The standard splitting lemma [21] states that

$$\forall (x,y) \in \mathcal{B}_\alpha \colon \Pr_{y' \xleftarrow{\$} \mathcal{Y}}[(x,y') \in \mathcal{B}] \geq \varepsilon - \alpha \tag{4}$$

$$\Pr_{(x,y) \xleftarrow{\$} \mathcal{B}}[(x,y) \in \mathcal{B}_\alpha] \geq \alpha/\varepsilon \tag{5}$$

For the conditional probability, we have that

$$\begin{aligned}
&\Pr_{y,y' \xleftarrow{\$} \mathcal{Y}, x \xleftarrow{\$} \mathcal{X}}[(x,y') \in \mathcal{B} \mid (x,y) \in \mathcal{B}] \\
&\geq \Pr_{y,y' \xleftarrow{\$} \mathcal{Y}, x \xleftarrow{\$} \mathcal{X}}[(x,y') \in \mathcal{B} \wedge (x,y) \in \mathcal{B}_\alpha \mid (x,y) \in \mathcal{B}] \\
&= \Pr_{y,y' \xleftarrow{\$} \mathcal{Y}, x \xleftarrow{\$} \mathcal{X}}[(x,y') \in \mathcal{B} \mid (x,y) \in \mathcal{B}_\alpha \cap \mathcal{B}] \cdot \Pr_{(x,y) \xleftarrow{\$} \mathcal{X} \times \mathcal{Y}}[(x,y) \in \mathcal{B}_\alpha \mid (x,y) \in \mathcal{B}] \\
&= \Pr_{y,y' \xleftarrow{\$} \mathcal{Y}, x \xleftarrow{\$} \mathcal{X}}[(x,y') \in \mathcal{B} \mid (x,y) \in \mathcal{B}_\alpha] \cdot \Pr_{(x,y) \xleftarrow{\$} \mathcal{X} \times \mathcal{Y}}[(x,y) \in \mathcal{B}_\alpha \mid (x,y) \in \mathcal{B}] \\
&= \Pr_{y,y' \xleftarrow{\$} \mathcal{Y}, x \xleftarrow{\$} \mathcal{X}}[(x,y') \in \mathcal{B} \mid (x,y) \in \mathcal{B}_\alpha] \cdot \Pr_{(x,y) \xleftarrow{\$} \mathcal{B}}[(x,y) \in \mathcal{B}_\alpha] \\
&\geq (\varepsilon - \alpha) \cdot \frac{\alpha}{\varepsilon},
\end{aligned}$$

where the inequalities follow from (4) and (5), respectively. We conclude the proof by

$$\begin{aligned}
&\Pr_{y,y' \xleftarrow{\$} \mathcal{Y}, x \xleftarrow{\$} \mathcal{X}}[(x,y') \in \mathcal{B} \wedge (x,y) \in \mathcal{B}] \\
&= \Pr_{y,y' \xleftarrow{\$} \mathcal{Y}, x \xleftarrow{\$} \mathcal{X}}[(x,y') \in \mathcal{B} \mid (x,y) \in \mathcal{B}] \cdot \Pr_{(x,y) \xleftarrow{\$} \mathcal{X} \times \mathcal{Y}}[(x,y) \in \mathcal{B}] \\
&\geq (\varepsilon - \alpha) \cdot \frac{\alpha}{\varepsilon} \cdot \varepsilon = (\varepsilon - \alpha) \cdot \alpha.
\end{aligned}$$

Lemma 5 (Subset Forking Lemma). *Fix any integer $Q \geq 1$ and a set $\mathcal{H}$ of size > 2 as well as a set of side outputs Σ, instances $\mathcal{I}$, and a randomness space Ω. Let C be an algorithm that on input $(I, \boldsymbol{h}) \in \mathcal{I} \times \mathcal{H}^Q$ and randomness $\omega \in \Omega$ returns a tuple (j, σ), where $1 \leq j \leq Q$ and $\sigma \in \Sigma$. We partition its input space $\mathcal{I} \times \Omega \times \mathcal{H}^Q$ into sets $\mathcal{W}_1, \ldots, \mathcal{W}_Q$ where for fixed $1 \leq j \leq Q$, $\mathcal{W}_j$ is the set of all $(I, \omega, \boldsymbol{h})$ that result in $(j, \sigma) \leftarrow \mathsf{C}(\boldsymbol{h}, I; \omega)$ for some arbitrary side output σ.*

For any $1 \leq j \leq Q$ and $\mathcal{B} \subseteq \mathcal{W}_j$ define

$$\mathtt{acc}(\mathcal{B}) := \Pr_{(I,\omega,\boldsymbol{h}) \xleftarrow{\$} \mathcal{I} \times \Omega \times \mathcal{H}^Q}[(I, \omega, \boldsymbol{h}) \in \mathcal{B}]$$

$$\mathtt{frk}(\mathcal{B}, j) := \Pr_{(I,\omega,\boldsymbol{h}) \xleftarrow{\$} \mathcal{I} \times \Omega \times \mathcal{H}^Q, \boldsymbol{h}' \xleftarrow{\$} \mathcal{C}^Q \vee | \boldsymbol{h}_{[j-1]}} \begin{bmatrix} \boldsymbol{h}_j \neq \boldsymbol{h}'_j \\ (I, \omega, \boldsymbol{h}) \in \mathcal{B} \wedge (I, \omega, \boldsymbol{h'}) \in \mathcal{B} \end{bmatrix}.$$

Then

$$\mathtt{frk}(\mathcal{B}, j) \geq \mathtt{acc}(\mathcal{B}) \cdot \left(\frac{\mathtt{acc}(\mathcal{B})}{4} - \frac{1}{|\mathcal{H}|} \right).$$

Proof. By applying Lemma 4 to $\varepsilon = \mathtt{acc}(B)$, $\alpha := \varepsilon/2$, and to the two sets $\mathcal{X} = \mathcal{I} \times \Omega \times \mathcal{H}^{j-1}$ and $\mathcal{Y} = \mathcal{H}^{Q-j+1}$, we obtain

$$\Pr_{(I,\omega,\boldsymbol{h}) \xleftarrow{\$} \mathcal{I}\times\Omega\times\mathcal{H}^Q, \boldsymbol{h}' \xleftarrow{\$} \mathcal{C}^Q \vee |\boldsymbol{h}_{[j-1]}}[(I,\omega,\boldsymbol{h}) \in \mathcal{B} \wedge (I,\omega,\boldsymbol{h}') \in \mathcal{B}] \geq \frac{\mathtt{acc}^2(\mathcal{B})}{4}.$$

Next, we observe that

$$\begin{aligned}
&\mathtt{frk}(\mathcal{B}, j) = \Pr[(I,\omega,\boldsymbol{h}) \in \mathcal{B} \wedge (I,\omega,\boldsymbol{h}') \in \mathcal{B} \wedge \boldsymbol{h}_j \neq \boldsymbol{h}'_j] \\
&= \Pr[(I,\omega,\boldsymbol{h}) \in \mathcal{B} \wedge (I,\omega,\boldsymbol{h}') \in \mathcal{B}] - \Pr[(I,\omega,\boldsymbol{h}) \in \mathcal{B} \wedge (I,\omega,\boldsymbol{h}') \in \mathcal{B} \wedge \boldsymbol{h}_j = \boldsymbol{h}'_j] \\
&\geq \Pr[(I,\omega,\boldsymbol{h}) \in \mathcal{B} \wedge (I,\omega,\boldsymbol{h}') \in \mathcal{B}] - \Pr[(I,\omega,\boldsymbol{h}) \in \mathcal{B} \wedge \boldsymbol{h}_j = \boldsymbol{h}'_j] \\
&= \Pr[(I,\omega,\boldsymbol{h}) \in \mathcal{B} \wedge (I,\omega,\boldsymbol{h}') \in \mathcal{B}] - \frac{\Pr[(I,\omega,\boldsymbol{h}) \in \mathcal{B}]}{|\mathcal{H}|},
\end{aligned}$$

where the last equation follows from independence and uniformity of $\boldsymbol{h}_j$ and $\boldsymbol{h}'_j$. We continue with

$$\begin{aligned}
&= \Pr[(I,\omega,\boldsymbol{h}) \in \mathcal{B} \wedge (I,\omega,\boldsymbol{h}') \in \mathcal{B}] - \frac{\Pr[(I,\omega,\boldsymbol{h}) \in \mathcal{B}]}{|\mathcal{H}|} \\
&\geq \frac{\mathtt{acc}^2(\mathcal{B})}{4} - \frac{\Pr[(I,\omega,\boldsymbol{h}) \in \mathcal{B}]}{|\mathcal{H}|} = \frac{\mathtt{acc}^2(\mathcal{B})}{4} - \frac{\mathtt{acc}(\mathcal{B})}{|\mathcal{H}|} \\
&= \mathtt{acc}(\mathcal{B}) \cdot \left(\frac{\mathtt{acc}(\mathcal{B})}{4} - \frac{1}{|\mathcal{H}|} \right),
\end{aligned}$$

which completes the proof.

Note that lemma 5 implies the version of the Forking Lemma in [6]. Namely, by, defining the set $\mathcal{W} = \bigcup_j \mathcal{W}_j$, $\mathtt{acc}(\mathcal{W}) = \Pr_{(I,\omega,\boldsymbol{h}) \xleftarrow{\$} \mathcal{I}\times\Omega\times\mathcal{H}^Q, (j,\sigma) \leftarrow \mathsf{C}(I,\boldsymbol{h};\omega)}[j \geq 1]$ and $\mathtt{frk} := \sum_{j=1}^{Q} \mathtt{frk}(\mathcal{W}_j, j)$, we obtain

$$\begin{aligned}
\mathtt{frk} &= \sum_{j=1}^{Q} \mathtt{frk}(\mathcal{W}_j, j) = \sum_{j=1}^{Q} \mathtt{acc}(\mathcal{W}_j) \cdot \left(\frac{\mathtt{acc}(\mathcal{W}_j)}{4} - \frac{1}{|\mathcal{H}|} \right) \\
&= \left(\sum_{j=1}^{Q} \frac{\mathtt{acc}^2(\mathcal{W}_j)}{4} \right) - \frac{\mathtt{acc}(\mathcal{W})}{|\mathcal{H}|} \geq \frac{1}{4Q} \left(\sum_{j=1}^{Q} \mathtt{acc}(\mathcal{W}_j) \right)^2 - \frac{\mathtt{acc}(\mathcal{W})}{|\mathcal{H}|} \\
&= \frac{1}{4Q} \mathtt{acc}^2(\mathcal{W}) - \frac{\mathtt{acc}(\mathcal{W})}{|\mathcal{H}|} = \mathtt{acc}(\mathcal{W}) \cdot \left(\frac{\mathtt{acc}(\mathcal{W})}{4Q} - \frac{1}{|\mathcal{H}|} \right),
\end{aligned}$$

where the inequality follows from Jensen's inequality (Lemma 3 in [6]).

6.3 Proof of Lemma 2

We will show in the following that for all $(I, \omega, \boldsymbol{h}) \xleftarrow{\$} (\mathcal{I} \times \Omega \times \mathcal{C}^{Q_\mathsf{V}}), d \in \mathcal{D}$:

$$\alpha_{i,j}(I, \omega, \boldsymbol{h}, d) := \Pr_{\boldsymbol{h'} \xleftarrow{\$} \mathcal{C}^{Q_\mathsf{V}} | \boldsymbol{h}_{[j-1]}} [\hat{\boldsymbol{\chi}}_i(I, \omega, \boldsymbol{h'}) \neq d \wedge \hat{\boldsymbol{J}}_i(I, \omega, \boldsymbol{h'}) = j]$$
$$\geq \mu_{i,j}(I, \omega, \boldsymbol{h})/2, \tag{6}$$

where

$$\mu_{i,j}(I, \omega, \boldsymbol{h}) := \Pr_{\boldsymbol{h'} \xleftarrow{\$} \mathcal{C}^{Q_\mathsf{V}} | \boldsymbol{h}_{[j-1]}} [(I, \omega, \boldsymbol{h'}) \in \mathcal{B}_{i,j} \wedge \boldsymbol{h}_j \neq \boldsymbol{h}'_j].$$

For a true/false statement s, define $B(s)$ as 1 if s is true and 0 otherwise. It is easy to see that (6) implies the theorem statement since

$$\begin{aligned}
\delta_{i,j} &= \Pr_{(I,\omega,\boldsymbol{h}) \xleftarrow{\$} (\mathcal{I} \times \Omega \times \mathcal{C}^{Q_\mathsf{V}}), \boldsymbol{h'} \xleftarrow{\$} \mathcal{C}^{Q_\mathsf{V}} | \boldsymbol{h}_{[j-1]}} \begin{bmatrix} \hat{\boldsymbol{\chi}}_i(I, \omega, \boldsymbol{h'}) \neq \hat{\boldsymbol{\chi}}_i(I, \omega, \boldsymbol{h}) \\ \wedge \hat{\boldsymbol{J}}_i(I, \omega, \boldsymbol{h}) = \hat{\boldsymbol{J}}_i(I, \omega, \boldsymbol{h'}) = j \end{bmatrix} \\
&= \sum_d \Pr_{(I,\omega,\boldsymbol{h}) \xleftarrow{\$} (\mathcal{I} \times \Omega \times \mathcal{C}^{Q_\mathsf{V}}), \boldsymbol{h'} \xleftarrow{\$} \mathcal{C}^{Q_\mathsf{V}} | \boldsymbol{h}_{[j-1]}} \begin{bmatrix} \hat{\boldsymbol{\chi}}_i(I, \omega, \boldsymbol{h'}) \neq d \wedge \hat{\boldsymbol{\chi}}_i(I, \omega, \boldsymbol{h}) = d \\ \wedge \hat{\boldsymbol{J}}_i(I, \omega, \boldsymbol{h}) = \hat{\boldsymbol{J}}_i(I, \omega, \boldsymbol{h'}) = j \end{bmatrix} \\
&= \sum_d \mathbf{E}_{I,\omega,\boldsymbol{h}}[B(\hat{\boldsymbol{\chi}}_i(I, \omega, \boldsymbol{h}) = d \wedge \hat{\boldsymbol{J}}_i(I, \omega, \boldsymbol{h}) = j) \cdot \alpha_{i,j}(I, \omega, \boldsymbol{h}, d)] \\
&\geq \frac{1}{2} \sum_d \mathbf{E}_{I,\omega,\boldsymbol{h}}[B(\hat{\boldsymbol{\chi}}_i(I, \omega, \boldsymbol{h}) = d \wedge \hat{\boldsymbol{J}}_i(I, \omega, \boldsymbol{h}) = j) \cdot \mu_{i,j}(I, \omega, \boldsymbol{h})],
\end{aligned}$$

where in the last step, we have applied linearity and monotonicity of the expectation and the fact that due to (6), for all $I, \omega, \boldsymbol{h} \in \mathcal{C}^{Q_\mathsf{V}}, d$, we have $\alpha_{i,j}(I, \omega, \boldsymbol{h}, d) \geq \mu_{i,j}(I, \omega, \boldsymbol{h})/2$. We continue with

$$\begin{aligned}
&\frac{1}{2} \sum_d \mathbf{E}_{I,\omega,\boldsymbol{h}}[B(\hat{\boldsymbol{\chi}}_i(I, \omega, \boldsymbol{h}) = d \wedge \hat{\boldsymbol{J}}_i(I, \omega, \boldsymbol{h}) = j) \cdot \mu_{i,j}(I, \omega, \boldsymbol{h})] \\
&= \frac{1}{2} \cdot \sum_d \Pr_{(I,\omega,\boldsymbol{h}) \xleftarrow{\$} (\mathcal{I} \times \Omega \times \mathcal{C}^{Q_\mathsf{V}}), \boldsymbol{h'} \xleftarrow{\$} \mathcal{C}^{Q_\mathsf{V}} | \boldsymbol{h}_{[j-1]}} \begin{bmatrix} \hat{\boldsymbol{\chi}}_i(I, \omega, \boldsymbol{h}) = d \wedge \hat{\boldsymbol{J}}_i(I, \omega, \boldsymbol{h}) = j \\ \wedge (I, \omega, \boldsymbol{h'}) \in \mathcal{B}_{i,j} \wedge \boldsymbol{h}_j \neq \boldsymbol{h}'_j \end{bmatrix} \\
&= \frac{1}{2} \cdot \Pr_{(I,\omega,\boldsymbol{h}) \xleftarrow{\$} (\mathcal{I} \times \Omega \times \mathcal{C}^{Q_\mathsf{V}}), \boldsymbol{h'} \xleftarrow{\$} \mathcal{C}^{Q_\mathsf{V}} | \boldsymbol{h}_{[j-1]}} \begin{bmatrix} \hat{\boldsymbol{J}}_i(I, \omega, \boldsymbol{h}) = j \\ \wedge (I, \omega, \boldsymbol{h'}) \in \mathcal{B}_{i,j} \wedge \boldsymbol{h}_j \neq \boldsymbol{h}'_j \end{bmatrix} && (7) \\
&\geq \frac{1}{2} \cdot \Pr_{(I,\omega,\boldsymbol{h}) \xleftarrow{\$} (\mathcal{I} \times \Omega \times \mathcal{C}^{Q_\mathsf{V}}), \boldsymbol{h'} \xleftarrow{\$} \mathcal{C}^{Q_\mathsf{V}} | \boldsymbol{h}_{[j-1]}} [(I, \omega, \boldsymbol{h}) \in \mathcal{B}_{i,j} \wedge (I, \omega, \boldsymbol{h'}) \in \mathcal{B}_{i,j} \wedge \boldsymbol{h}_j \neq \boldsymbol{h}'_j] \\
& && (8) \\
&= \frac{1}{2} \cdot \mathtt{frk}(\mathcal{B}_{i,j}, j) && (9) \\
&\geq \beta_{i,j} \left(\beta_{i,j}/8 - \frac{1}{2q} \right), && (10)
\end{aligned}$$

where from (7) to (8), we have used the fact that $(I, \omega, \boldsymbol{h'}) \in \mathcal{B}_{i,j}$ implies $\hat{\boldsymbol{J}}_i(I, \omega, \boldsymbol{h'}) = j$. The inequality from (9) to (10) follows directly from Lemma 5.

We prove (6) by analyzing two cases. For all $I, \omega, \boldsymbol{h}, d$, we define

$$\gamma_{i,j}(I,\omega,\boldsymbol{h},d) := \Pr_{\boldsymbol{h'} \xleftarrow{\$} \mathcal{C}^{Q_\mathsf{V}}|\boldsymbol{h}_{[j-1]}} [\hat{\boldsymbol{\chi}}_i(I,\omega,\boldsymbol{h'}) = d \wedge (I,\omega,\boldsymbol{h'}) \in \mathcal{B}_{i,j} \wedge h_j \neq h'_j].$$

Case 1: $\gamma_{i,j}(I,\omega,\boldsymbol{h},d) \geq \mu_{i,j}(I,\omega,\boldsymbol{h})/2$.

Note that in this case we can assume $d \neq C_i(I,\omega,\boldsymbol{h})$. (This is because if $d = C_i(I,\omega,\boldsymbol{h})$, then $\gamma_{i,j}(I,\omega,\boldsymbol{h},d) \leq \Pr[\hat{\boldsymbol{\chi}}_i(I,\omega,\boldsymbol{h'}) = C_i(I,\omega,\boldsymbol{h}) \wedge (I,\omega,\boldsymbol{h'}) \in \mathcal{B}_{i,j}] = 0$ which would trivialize the claim.)

$$\begin{aligned}
\alpha_{i,j}(I,\omega,\boldsymbol{h},d) &= \Pr_{\boldsymbol{h'} \xleftarrow{\$} \mathcal{C}^{Q_\mathsf{V}}|\boldsymbol{h}_{[j-1]}} [\hat{\boldsymbol{\chi}}_i(I,\omega,\boldsymbol{h'}) \neq d \wedge \hat{\boldsymbol{J}}_i(I,\omega,\boldsymbol{h'}) = j] \\
&\geq \Pr[\hat{\boldsymbol{\chi}}_i(I,\omega,\boldsymbol{h'}) = C_i(I,\omega,\boldsymbol{h}) \wedge \hat{\boldsymbol{J}}_i(I,\omega,\boldsymbol{h'}) = j] \\
&\geq \Pr[\hat{\boldsymbol{\chi}}_i(I,\omega,\boldsymbol{h'}) = d \wedge \hat{\boldsymbol{J}}_i(I,\omega,\boldsymbol{h'}) = j]
\end{aligned}$$

Using again that $(I,\omega,\boldsymbol{h'}) \in \mathcal{B}_{i,j}$ implies $\hat{\boldsymbol{J}}_i(I,\omega,\boldsymbol{h'}) = j$, we obtain

$$\begin{aligned}
&\Pr[\hat{\boldsymbol{\chi}}_i(I,\omega,\boldsymbol{h'}) = d \wedge \hat{\boldsymbol{J}}_i(I,\omega,\boldsymbol{h'}) = j] \geq \Pr[\hat{\boldsymbol{\chi}}_i(I,\omega,\boldsymbol{h'}) = d \wedge (I,\omega,\boldsymbol{h'}) \in \mathcal{B}_{i,j}] \\
&\geq \gamma_{i,j}(I,\omega,\boldsymbol{h},d) \geq \mu_{i,j}(I,\omega,\boldsymbol{h})/2.
\end{aligned}$$

Case 2: $\gamma_{i,j}(I,\omega,\boldsymbol{h},d) < \mu_{i,j}(I,\omega,\boldsymbol{h})/2$. Now,

$$\begin{aligned}
\alpha_{i,j}(I,\omega,\boldsymbol{h},d) &= \Pr_{\boldsymbol{h'} \xleftarrow{\$} \mathcal{C}^{Q_\mathsf{V}}|\boldsymbol{h}_{[j-1]}} [\hat{\boldsymbol{\chi}}_i(I,\omega,\boldsymbol{h'}) \neq d \wedge \hat{\boldsymbol{J}}_i(I,\omega,\boldsymbol{h'}) = j] \\
&\geq \Pr[\hat{\boldsymbol{\chi}}_i(I,\omega,\boldsymbol{h'}) \neq d \wedge (I,\omega,\boldsymbol{h'}) \in \mathcal{B}_{i,j} \wedge \boldsymbol{h}_j \neq \boldsymbol{h}'_j] \\
&= \Pr[(I,\omega,\boldsymbol{h'}) \in \mathcal{B}_{i,j} \wedge \boldsymbol{h}_j \neq \boldsymbol{h}'_j] \\
&\quad - \Pr[\hat{\boldsymbol{\chi}}_i(I,\omega,\boldsymbol{h'}) = d \wedge (I,\omega,\boldsymbol{h'}) \in \mathcal{B}_{i,j} \wedge \boldsymbol{h}_j \neq \boldsymbol{h}'_j] \\
&= \mu_{i,j}(I,\omega,\boldsymbol{h}) - \gamma_{i,j}(I,\omega,\boldsymbol{h},d) > \mu_{i,j}(I,\omega,\boldsymbol{h})/2.
\end{aligned}$$

This proves (6) and hence the lemma.

6.4 Proof of Lemma 3

Consider again the algorithm A_i in Figure 11 and its internal variables. On input $(I = sk, \omega = (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{h})$, A_i invokes M on $pk = \mathsf{F}(sk)$ and randomness ω_M and answers its queries using the values in $\boldsymbol{r}, \boldsymbol{h}$. Similarly as before, this allows us to fix an execution of M (within A_i) via a tuple of the form $(I,\omega,\boldsymbol{h}) = (I,(\omega_\mathsf{M},\boldsymbol{r}),\boldsymbol{h})$. Let $\boldsymbol{c}(I,\omega,\boldsymbol{h})$ denote the vector of challenge values as defined in Line 20 of Figure 11.

Recall that we have assumed that $\mathsf{F} : \mathcal{D} \longrightarrow \mathcal{R}$ and the existence of a torsion-free element $z^* \in \mathcal{D} \setminus \{0\}$ such that (i) $\mathsf{F}(z^*) = 0$; and (ii) $\forall s \in \mathcal{C} : s \cdot z^* = 0 \implies s = 0$.

Lemma 6. *Consider the mapping*

$$\Phi : \mathcal{W} \longrightarrow (\mathcal{I} \times \Omega \times \mathcal{C}^{Q_{\mathsf{V}}}), \quad (sk, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) \mapsto (sk + z^*, (\omega_{\mathsf{M}}, \boldsymbol{r} - z^* \cdot \boldsymbol{c}(I, \omega, \boldsymbol{h})), \boldsymbol{h}),$$

where we make the convention that for $v \in \mathcal{D} \cup \mathcal{C} \cup \mathcal{R}, v \cdot \perp := 0$. Then Φ is a permutation on $\mathcal{W}$.

For the proof we require the following lemma.

Lemma 7. *Let $(I, \omega, \boldsymbol{h}) \in \mathcal{W}$. Then the tuples $(I, \omega, \boldsymbol{h})$ and $\Phi(I, \omega, \boldsymbol{h})$ fix the same execution of* M.

Proof. We show that M sees identical values in both executions corresponding to $(I, \omega, \boldsymbol{h})$ and $\Phi(I, \omega, \boldsymbol{h})$. To this end we consider all values in the view of M.

- **Initial input to M.** Since Φ does not alter the values of ω_{M}, we only need to verify that M obtains the same public key in both executions. This is ensured via $\mathsf{F}(sk + z^*) = \mathsf{F}(sk) + \mathsf{F}(z^*) = \mathsf{F}(sk) = pk$
- **Outputs of oracle $\mathtt{P}_1$.** Oracle $\mathtt{P}_1$ consecutively returns the values from $\boldsymbol{R} = \mathsf{F}(\boldsymbol{r})$, as defined in Line 01 of Figure 11. They remain the same in both executions since $\mathsf{F}(\boldsymbol{r}) = \boldsymbol{R} = \boldsymbol{R} - 0 \cdot \boldsymbol{c}(I, \omega, \boldsymbol{h}) = \mathsf{F}(\boldsymbol{r}) - \mathsf{F}(z^*) \cdot \boldsymbol{c}(I, \omega, \boldsymbol{h}) = \mathsf{F}(\boldsymbol{r} - z^* \cdot \boldsymbol{c}(I, \omega, \boldsymbol{h}))$.
- **Outputs of oracle $\mathtt{P}_2$.** Oracle $\mathtt{P}_2$ consecutively returns the values from $\boldsymbol{s} = \boldsymbol{c}sk + \boldsymbol{r}$, as defined in Line 19 of Figure 11. They remain the same in both executions since $\boldsymbol{r} + sk \cdot \boldsymbol{c}(I, \omega, \boldsymbol{h}) = \boldsymbol{s} = \boldsymbol{r} - z^* \cdot \boldsymbol{c}(I, \omega, \boldsymbol{h}) + z^* \cdot \boldsymbol{c}(I, \omega, \boldsymbol{h}) + sk \cdot \boldsymbol{c}(I, \omega, \boldsymbol{h}) = (\boldsymbol{r} - z^* \cdot \boldsymbol{c}(I, \omega, \boldsymbol{h})) + (sk + z^*) \cdot \boldsymbol{c}(I, \omega, \boldsymbol{h})$.
- **Outputs of oracle $\mathtt{V}_2$.** Oracle $\mathtt{P}_2$ consecutively returns the values from $\boldsymbol{b}$. They remain the same in both executions since they depend on $\boldsymbol{R}$, $\boldsymbol{h}$, and the randomness ω_{M}.

Thus, $(I, \omega, \boldsymbol{h})$ and $\Phi(I, \omega, \boldsymbol{h})$ fix the same executions of M.

Proof (Proof of Lemma 6). First note that Lemma 7 implies that Φ maps to $\mathcal{W}$. It remains to prove that Φ is also a bijection. Suppose Φ is not injective. Thus, for distinct tuples $(I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) \neq (I', (\omega'_{\mathsf{M}}, \boldsymbol{r'}), \boldsymbol{h'})$, $\Phi(I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) = \Phi(I', (\omega'_{\mathsf{M}}, \boldsymbol{r'}), \boldsymbol{h'})$. This implies $\omega_{\mathsf{M}} = \omega'_{\mathsf{M}}$ and $\boldsymbol{h} = \boldsymbol{h'}$. Similarly, $sk + z^* = sk' + z^*$, which implies that $sk = sk'$. Lastly, $\boldsymbol{r} - z^* \cdot \boldsymbol{c}(I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) = \boldsymbol{r'} - z^* \cdot \boldsymbol{c}(I', \omega'_{\mathsf{M}}, \boldsymbol{r'}, \boldsymbol{h'})$. Since $\Phi(I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) = \Phi(I', (\omega'_{\mathsf{M}}, \boldsymbol{r'}), \boldsymbol{h'})$, by Claim 7, $(I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h})$ and $(I', (\omega'_{\mathsf{M}}, \boldsymbol{r'}), \boldsymbol{h'})$ fix the same execution and therefore also $\boldsymbol{c}(I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) = \boldsymbol{c}(I', (\omega'_{\mathsf{M}}, \boldsymbol{r'}), \boldsymbol{h'})$. This implies $\boldsymbol{r} = \boldsymbol{r'}$, leading to the contradiction $(I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) = (I', (\omega'_{\mathsf{M}}, \boldsymbol{r'}), \boldsymbol{h'})$.

To prove that Φ is surjective, we consider the function $\Phi^{-1} : (\mathcal{I} \times \Omega \times \mathcal{C}^{Q_{\mathsf{V}}}) \longrightarrow (\mathcal{I} \times \Omega \times \mathcal{C}^{Q_{\mathsf{V}}})$, defined as $\Phi^{-1}(sk, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) = (sk - z^*, (\omega_{\mathsf{M}}, \boldsymbol{r} + z^* \cdot \boldsymbol{c}(I, \omega, \boldsymbol{h})), \boldsymbol{h})$, which is the inverse of Φ. With the same argument as above, one can also prove that Φ^{-1} is injective which implies the surjectivity of Φ.

We now introduce the following notation. Let $\mathcal{B} = \bigcup_{i,j} \mathcal{B}_{i,j}$ and let $\mathcal{G} = \mathcal{W} \setminus \mathcal{B}$. That is, for all $(I, \omega, \boldsymbol{h}) \in \mathcal{G}$, we have $\forall k \in [Q_{\mathsf{P}_2} + 1] : \hat{\boldsymbol{\chi}}_k(I, \omega, \boldsymbol{h}) = C_k(I, \omega, \boldsymbol{h})$.

The following combinatorial lemma lower bounds the probability that $\hat{\boldsymbol{\chi}}$ takes different values (i.e., differs in at least one component) as a result of distinct instances $I = sk, I' = sk + z^*$.

Lemma 8. *For any fixed* $(I, (\omega_{\mathsf{M}}, \boldsymbol{r})) \in \mathcal{I} \times \Omega$,

$$\Pr_{\boldsymbol{h} \xleftarrow{\$} \mathcal{C}^{Q_{\mathsf{V}}}} [(I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) \in \mathcal{G} \wedge \Phi(I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) \in \mathcal{G}] \leq \frac{Q_{\mathsf{V}}^{Q_{\mathsf{P}_2}+1} \cdot \binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{q}.$$

Proof. We argue by contradiction. Thus, assume that for some $(I, (\omega_{\mathsf{M}}, \boldsymbol{r})) \in \mathcal{I} \times \Omega$,

$$\Pr_{\boldsymbol{h} \xleftarrow{\$} \mathcal{C}^{Q_{\mathsf{V}}}} [(I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) \in \mathcal{G} \wedge \Phi(I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) \in \mathcal{G}] > \frac{Q_{\mathsf{V}}^{Q_{\mathsf{P}_2}+1} \cdot \binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{q}.$$

Then there exist a set $\{u_1, ..., u_{Q_{\mathsf{P}_2}+1}\}$ of $Q_{\mathsf{P}_2} + 1$ distinct indices from $[Q_{\mathsf{V}}]$ such that

$$\Pr_{\boldsymbol{h} \xleftarrow{\$} \mathcal{C}^{Q_{\mathsf{V}}}} \begin{bmatrix} ((I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) \in \mathcal{G}) \wedge \ (\Phi(I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) \in \mathcal{G}) \\ \wedge \forall j : \hat{\boldsymbol{J}}_j (I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) = u_j \end{bmatrix} > \frac{\binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{q}.$$

Similarly, there exists a vector $\boldsymbol{d} \in (\mathcal{C} \cup \{\perp\})^{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}$ of challenges such that $\boldsymbol{d}$ has exactly Q_{P_1} entries which are $\perp$ and furthermore has the property that

$$\Pr_{\boldsymbol{h} \xleftarrow{\$} \mathcal{C}^{Q_{\mathsf{V}}}} \begin{bmatrix} ((I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) \in \mathcal{G}) \wedge \ (\Phi(I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) \in \mathcal{G}) \\ \wedge (\boldsymbol{c}(I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) = \boldsymbol{d}) \wedge \left(\forall j : \hat{\boldsymbol{J}}_j (I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) = u_j\right) \end{bmatrix} > \frac{1}{q^{Q_{\mathsf{P}_2}+1}}.$$

Lastly, there exists a set $\{v_1, ..., v_{Q_{\mathsf{V}}-Q_{\mathsf{P}_2}-1}\}$ of $Q_{\mathsf{V}} - Q_{\mathsf{P}_2} - 1$ distinct indices from $[Q_{\mathsf{V}}] \setminus \{u_1, ..., u_{Q_{\mathsf{P}_2}+1}\}$ and a vector $(\tilde{\boldsymbol{h}}_{v_1}, ..., \tilde{\boldsymbol{h}}_{v_{Q_{\mathsf{V}}-Q_{\mathsf{P}_2}-1}}) \in \mathcal{C}^{Q_{\mathsf{V}}-Q_{\mathsf{P}_2}-1}$ such that

$$\Pr_{\boldsymbol{h} \xleftarrow{\$} \mathcal{C}^{Q_{\mathsf{V}}}} \begin{bmatrix} ((I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) \in \mathcal{G}) \wedge (\Phi(I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) \in \mathcal{G}) \wedge \ (\boldsymbol{c}(I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) = \boldsymbol{d}) \\ \wedge \left(\forall j : (\hat{\boldsymbol{J}}_j (I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) = u_j\right) \wedge \left(\forall j : \boldsymbol{h}_{v_j} = \tilde{\boldsymbol{h}}_{v_j}\right) \end{bmatrix}$$
$$> \frac{1}{q^{Q_{\mathsf{P}_2}+1} q^{Q_{\mathsf{V}}-Q_{\mathsf{P}_2}-1}} = \frac{1}{q^{Q_{\mathsf{V}}}}.$$

Since the random variable $\boldsymbol{h}$ takes a particular value $\boldsymbol{k} \in \mathcal{C}^{Q_{\mathsf{V}}}$ with probability exactly $q^{-Q_{\mathsf{V}}}$, the statement inside the probability term above must be true for at least two distinct vectors $\boldsymbol{k}, \boldsymbol{k'} \in \mathcal{C}^{Q_{\mathsf{V}}}$. Furthermore, since the condition in the probability term above fixes all but the $Q_{\mathsf{P}_2} + 1$ components $\{u_1, ..., u_{Q_{\mathsf{P}_2}+1}\}$ of $\boldsymbol{k}$ and $\boldsymbol{k'}$, there exists an index $i \in [Q_{\mathsf{P}_2} + 1]$ s.t. $\boldsymbol{k}_{u_i} \neq \boldsymbol{k}'_{u_i}$.

W.l.o.g., let i be the smallest such index. This implies that $\forall j < u_i : \boldsymbol{k}_j = \boldsymbol{k}'_j$ and $\boldsymbol{k}_{u_i} \neq \boldsymbol{k}'_{u_i}$. Therefore,

$$C_i(I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{k}) = C_i(I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{k'}). \tag{11}$$

Furthermore, by Lemma 7,

$$\begin{aligned}
&C_i(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k}) = \hat{\boldsymbol{s}}'_i(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k}) - sk\cdot\boldsymbol{k}_{u_i}\\
&= \hat{\boldsymbol{s}}'_i(\Phi(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k})) - sk\cdot\boldsymbol{k}_{u_i}\\
&= \hat{\boldsymbol{s}}'_i(\Phi(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k})) - sk\cdot\boldsymbol{k}_{u_i} + z^*\cdot\boldsymbol{k}_{u_i} - z^*\cdot\boldsymbol{k}_{u_i}\\
&= \hat{\boldsymbol{s}}'_i(\Phi(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k})) - (sk+z^*)\cdot\boldsymbol{k}_{u_i} + z^*\cdot\boldsymbol{k}_{u_i}\\
&= C_i(\Phi(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k})) + z^*\cdot\boldsymbol{k}_{u_i}\\
&= C_i(I,\omega_{\mathsf{M}},\boldsymbol{r} - z^*\cdot\boldsymbol{c}(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k}),\boldsymbol{k}) + z^*\cdot\boldsymbol{k}_{u_i}. \qquad (12)
\end{aligned}$$

Analogously, we infer

$$\begin{aligned}
&C_i(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k}') = \hat{\boldsymbol{s}}'_i(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k}') - sk\cdot\boldsymbol{k}'_{u_i}\\
&= C_i(I,\omega_{\mathsf{M}},\boldsymbol{r} - z^*\cdot\boldsymbol{c}(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k}'),\boldsymbol{k}') + z^*\cdot\boldsymbol{k}'_{u_i}. \qquad (13)
\end{aligned}$$

Combining (in this order) equations 12, 11, and 13, we obtain:

$$\begin{aligned}
&C_i(I,\omega_{\mathsf{M}},\boldsymbol{r} - z^*\cdot\boldsymbol{c}(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k}),\boldsymbol{k}) + z^*\cdot\boldsymbol{k}_{u_i}\\
&= C_i(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k}) = C_i(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k}')\\
&= C_i(I,\omega_{\mathsf{M}},\boldsymbol{r} - z^*\cdot\boldsymbol{c}(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k}'),\boldsymbol{k}') + z^*\cdot\boldsymbol{k}'_{u_i}. \qquad (14)
\end{aligned}$$

Since above we have fixed $\boldsymbol{c}(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k}) = \boldsymbol{c}(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k}') = \boldsymbol{d}$, we also know that

$$\begin{aligned}
&C_i(I,\omega_{\mathsf{M}},\boldsymbol{r} - z^*\cdot\boldsymbol{c}(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k}),\boldsymbol{k})\\
&= C_i(I,\omega_{\mathsf{M}},\boldsymbol{r} - z^*\cdot\boldsymbol{d},\boldsymbol{k})\\
&= C_i(I,\omega_{\mathsf{M}},\boldsymbol{r} - z^*\cdot\boldsymbol{d},\boldsymbol{k}') \qquad (15)\\
&= C_i(I,\omega_{\mathsf{M}},\boldsymbol{r} - z^*\cdot\boldsymbol{c}(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k}'),\boldsymbol{k}'), \qquad (16)
\end{aligned}$$

where 15 follows again from the fact that $\forall j < u_i : \boldsymbol{k}_j = \boldsymbol{k}'_j$. By combining 14 and 16, it now follows that $z^*\cdot\boldsymbol{k}_{u_i} = z^*\cdot\boldsymbol{k}'_{u_i}$ or, equivalently, $z^*\cdot(\boldsymbol{k}_{u_i} - \boldsymbol{k}'_{u_i}) = 0$. Thus, torsion-freeness of z^* implies that $\boldsymbol{k}_{u_i} = \boldsymbol{k}'_{u_i}$ which contradicts the assumption that $\boldsymbol{k}_{u_i} \neq \boldsymbol{k}'_{u_i}$. This completes the proof.

Corollary 2. $\Pr_{(I,\omega,\boldsymbol{h})\xleftarrow{\$}(\mathcal{I}\times\Omega\times\mathcal{C}^{Q_{\mathsf{V}}})}[(I,\omega,\boldsymbol{h})\in\mathcal{G}\wedge\Phi(I,\omega,\boldsymbol{h})\in\mathcal{G}] \leq \frac{Q_{\mathsf{V}}^{Q_{\mathsf{P}_2}+1}\cdot\binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{q}.$

Discussion. The lower bound in Corollary 2 exponentially depreciates with the number Q_{P_2} of parallel sessions allowed in the **OMMIM** experiment. Unfortunately, the ROS-attack in 4.2 shows that the bound in Corollary 2 can not be improved beyond a factor of $\binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}$. The reason for this is that our attacker computes $\hat{\boldsymbol{\chi}}$ in a manner that does not depend on $\boldsymbol{h}$, but only on ω, I (more precisely, any contribution of $\boldsymbol{h}$ 'cancels out' in the values returned by the attacker). Therefore, $\hat{\boldsymbol{\chi}}$ *always* takes the 'most likely' value according to 3 in the sense that, regardless of $\boldsymbol{h}$, the attacker can force $(\omega, I, \boldsymbol{h}) \in \mathcal{G}$ *and* $\Phi(\omega, I, \boldsymbol{h}) \in \mathcal{G}$.

Lemma 9. $\Pr_{(I,\omega,\boldsymbol{h}) \xleftarrow{\$} (\mathcal{I}\times\Omega\times\mathcal{C}^{Q_\mathsf{V}})}[(I,\omega,\boldsymbol{h}) \in \mathcal{B}] \geq \frac{1}{2}\left(\varepsilon - \frac{Q_\mathsf{V}^{Q_{\mathsf{P}_2}+1}\cdot\binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{q}\right)$.

Proof. We partition $\mathcal{G}$ into subsets $\mathcal{G}_g, \mathcal{G}_b$ such that all elements in $\mathcal{G}_g$ are mapped into $\mathcal{G}$ via Φ and all elements in $\mathcal{G}_b$ are mapped into $\mathcal{B}$ via Φ. It follows that

$$\begin{aligned}&\Pr_{(I,\omega,\boldsymbol{h}) \xleftarrow{\$} (\mathcal{I}\times\Omega\times\mathcal{C}^{Q_\mathsf{V}})}[(I,\omega,\boldsymbol{h}) \in \mathcal{G}]\\ &= \Pr_{(I,\omega,\boldsymbol{h}) \xleftarrow{\$} (\mathcal{I}\times\Omega\times\mathcal{C}^{Q_\mathsf{V}})}[(I,\omega,\boldsymbol{h}) \in \mathcal{G}_g] + \Pr_{(I,\omega,\boldsymbol{h}) \xleftarrow{\$} (\mathcal{I}\times\Omega\times\mathcal{C}^{Q_\mathsf{V}})}[(I,\omega,\boldsymbol{h}) \in \mathcal{G}_b]. \quad (17)\end{aligned}$$

By Corollary 2 and because Φ is a bijection, we can infer that

$$\Pr_{(I,\omega,\boldsymbol{h}) \xleftarrow{\$} (\mathcal{I}\times\Omega\times\mathcal{C}^{Q_\mathsf{V}})}[(I,\omega,\boldsymbol{h}) \in \mathcal{G}_g] \leq \frac{Q_\mathsf{V}^{Q_{\mathsf{P}_2}+1}\cdot\binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{q}, \quad (18)$$

$$\Pr_{(I,\omega,\boldsymbol{h}) \xleftarrow{\$} (\mathcal{I}\times\Omega\times\mathcal{C}^{Q_\mathsf{V}})}[(I,\omega,\boldsymbol{h}) \in \mathcal{G}_b] \leq \Pr_{(I,\omega,\boldsymbol{h}) \xleftarrow{\$} (\mathcal{I}\times\Omega\times\mathcal{C}^{Q_\mathsf{V}})}[(I,\omega,\boldsymbol{h}) \in \mathcal{B}]. \quad (19)$$

It follows from 17,18, 19 that

$$\Pr[(I,\omega,\boldsymbol{h}) \in \mathcal{G}] \leq \frac{Q_\mathsf{V}^{Q_{\mathsf{P}_2}+1}\cdot\binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{q} + \Pr[(I,\omega,\boldsymbol{h}) \in \mathcal{B}]. \quad (20)$$

From 20, we can bound $\Pr[(I,\omega,\boldsymbol{h}) \in \mathcal{B}]$ as

$$\begin{aligned}\Pr[(I,\omega,\boldsymbol{h}) \in \mathcal{B}] &= \Pr[(I,\omega,\boldsymbol{h}) \in \mathcal{W}] - \Pr[(I,\omega,\boldsymbol{h}) \in \mathcal{G}]\\ &\geq \Pr[(I,\omega,\boldsymbol{h}) \in \mathcal{W}] - \Pr[(I,\omega,\boldsymbol{h}) \in \mathcal{B}] - \frac{Q_\mathsf{V}^{Q_{\mathsf{P}_2}+1}\cdot\binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{q}.\end{aligned}$$

Since $\varepsilon = \Pr_{(I,\omega,\boldsymbol{h}) \xleftarrow{\$} (\mathcal{I}\times\Omega\times\mathcal{C}^{Q_\mathsf{V}})}[(I,\omega,\boldsymbol{h}) \in \mathcal{W}]$, we finally obtain

$$\Pr_{(I,\omega,\boldsymbol{h}) \xleftarrow{\$} (\mathcal{I}\times\Omega\times\mathcal{C}^{Q_\mathsf{V}})}[(I,\omega,\boldsymbol{h}) \in \mathcal{B}] \geq \frac{1}{2}\left(\varepsilon - \frac{Q_\mathsf{V}^{Q_{\mathsf{P}_2}+1}\cdot\binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{q}\right).$$

We are now ready to prove Lemma 3, i.e., we show that there exist $i \in [Q_{\mathsf{P}_2}+1], j \in [Q_\mathsf{V}]$ such that $\beta_{i,j} > \left(\varepsilon - \frac{Q_\mathsf{V}^{Q_{\mathsf{P}_2}+1}\cdot\binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{q}\right)\cdot\frac{1}{2Q_\mathsf{V}(Q_{\mathsf{P}_2}+1)}$. Toward a contradiction, suppose instead that for all $i \in [Q_{\mathsf{P}_2}+1], j \in [Q_\mathsf{V}]$, we have that

$$\Pr_{(I,\omega,\boldsymbol{h}) \xleftarrow{\$} (\mathcal{I}\times\Omega\times\mathcal{C}^{Q_\mathsf{V}})}[(I,\omega,\boldsymbol{h}) \in \mathcal{B}_{i,j}] < \left(\varepsilon - \frac{Q_\mathsf{V}^{Q_{\mathsf{P}_2}+1}\cdot\binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{q}\right)\cdot\frac{1}{2Q_\mathsf{V}(Q_{\mathsf{P}_2}+1)}.$$

By Lemma 9,

$$\frac{1}{2}\left(\varepsilon - \frac{Q_{\mathsf{V}}^{Q_{\mathsf{P}_2}+1} \cdot \binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{q}\right) \leq \Pr[(I,\omega,\boldsymbol{h}) \in \mathcal{B}] = \Pr[(I,\omega,\boldsymbol{h}) \in \bigcup_{i,j} \mathcal{B}_{i,j}]$$

$$\leq \sum_{i,j} \Pr[(I,\omega,\boldsymbol{h}) \in \mathcal{B}_{i,j}] < \frac{1}{2}\left(\varepsilon - \frac{Q_{\mathsf{V}}^{Q_{\mathsf{P}_2}+1} \cdot \binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{q}\right).$$

This is a contradiction.

References

1. M. Abdalla, J. H. An, M. Bellare, and C. Namprempre. From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security. In L. R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 418–433. Springer, Heidelberg, Apr. / May 2002. 2, 8
2. M. Backendal, M. Bellare, J. Sorrell, and J. Sun. The fiat-shamir zoo: Relating the security of different signature variants. Cryptology ePrint Archive, Report 2018/775, 2018. `https://eprint.iacr.org/2018/775`. 2, 5, 6, 10
3. F. Baldimtsi and A. Lysyanskaya. Anonymous credentials light. In A.-R. Sadeghi, V. D. Gligor, and M. Yung, editors, *ACM CCS 13*, pages 1087–1098. ACM Press, Nov. 2013. 1
4. F. Baldimtsi and A. Lysyanskaya. On the security of one-witness blind signature schemes. In K. Sako and P. Sarkar, editors, *ASIACRYPT 2013, Part II*, volume 8270 of *LNCS*, pages 82–99. Springer, Heidelberg, Dec. 2013. 1
5. M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, and H. Shacham. Randomizable proofs and delegatable anonymous credentials. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 108–125. Springer, Heidelberg, Aug. 2009. 1
6. M. Bellare and G. Neven. Multi-signatures in the plain public-key model and a general forking lemma. In A. Juels, R. N. Wright, and S. Vimercati, editors, *ACM CCS 06*, pages 390–399. ACM Press, Oct. / Nov. 2006. 3, 4, 18, 23
7. M. Bellare and A. Palacio. GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In M. Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 162–177. Springer, Heidelberg, Aug. 2002. 2, 8
8. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73. ACM Press, Nov. 1993. 1
9. S. Brands. Untraceable off-line cash in wallets with observers (extended abstract). In D. R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 302–318. Springer, Heidelberg, Aug. 1994. 1
10. J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Compact e-cash. In R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 302–321. Springer, Heidelberg, May 2005. 1
11. J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In B. Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer, Heidelberg, May 2001. 1

12. D. Chaum. Blind signatures for untraceable payments. In D. Chaum, R. L. Rivest, and A. T. Sherman, editors, *CRYPTO'82*, pages 199–203. Plenum Press, New York, USA, 1982. 1
13. D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. In S. Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 319–327. Springer, Heidelberg, Aug. 1990. 1
14. U. Feige, A. Fiat, and A. Shamir. Zero-knowledge proofs of identity. *Journal of Cryptology*, 1(2):77–94, June 1988. 2
15. R. Gennaro. Multi-trapdoor commitments and their applications to proofs of knowledge secure under concurrent man-in-the-middle attacks. In M. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 220–236. Springer, Heidelberg, Aug. 2004. 2
16. V. Lyubashevsky. Lattice-based identification schemes secure under active attacks. In R. Cramer, editor, *PKC 2008*, volume 4939 of *LNCS*, pages 162–179. Springer, Heidelberg, Mar. 2008. 4
17. L. Minder and A. Sinclair. The extended k-tree algorithm. In C. Mathieu, editor, *20th SODA*, pages 586–595. ACM-SIAM, Jan. 2009. 6
18. T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In E. F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 31–53. Springer, Heidelberg, Aug. 1993. 1, 2
19. T. Okamoto and K. Ohta. Universal electronic cash. In J. Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 324–337. Springer, Heidelberg, Aug. 1992. 1
20. D. Pointcheval and J. Stern. New blind signatures equivalent to factorization (extended abstract). In *ACM CCS 97*, pages 92–99. ACM Press, Apr. 1997. 1, 2
21. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, June 2000. 1, 3, 4, 17, 22
22. F. Rodriguez-Henriquez, D. Ortiz-Arroyo, and C. Garcia-Zamora. Yet another improvement over the mu-varadharajan e-voting protocol. *Comput. Stand. Interfaces*, 29(4):471–480, 2007. 1
23. M. Rückert. Lattice-based blind signatures. In M. Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 413–430. Springer, Heidelberg, Dec. 2010. 1, 4
24. C.-P. Schnorr. Security of blind discrete log signatures against interactive attacks. In S. Qing, T. Okamoto, and J. Zhou, editors, *ICICS 01*, volume 2229 of *LNCS*, pages 1–12. Springer, Heidelberg, Nov. 2001. 3, 6
25. D. Wagner. A generalized birthday problem. In M. Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 288–303. Springer, Heidelberg, Aug. 2002. 3, 6

Acknowledgments

We would like to thank David Pointcheval for helpful discussions and for answering many of our questions.