Lattice-Based Blind Signatures, Revisited

Eduard Hauck[1], Eike Kiltz[1], Julian Loss[2], and Ngoc Khanh Nguyen[3]

[1] Ruhr-Universität Bochum, Bochum, Germany
{eduard.hauck,eike.kiltz}@rub.de
[2] University of Maryland, College Park, USA
julian.loss@gmail.com
[3] ETH Zurich; IBM Research, Zurich, Switzerland
NKN@zurich.ibm.com

Abstract. We observe that all previously known lattice-based blind signature schemes contain subtle flaws in their security proofs (e.g., Rückert, ASIACRYPT '08) or can be attacked (e.g., BLAZE by Alkadri et al., FC '20). Motivated by this, we revisit the problem of constructing blind signatures from standard lattice assumptions.

We propose a new three-round lattice-based blind signature scheme whose security can be proved, in the random oracle model, from the standard SIS assumption. Our starting point is a modified version of the (insecure) BLAZE scheme, which itself is based Lyubashevsky's three-round identification scheme combined with a new aborting technique to reduce the correctness error. Our proof builds upon and extends the recent modular framework for blind signatures of Hauck, Kiltz, and Loss (EUROCRYPT '19). It also introduces several new techniques to overcome the additional challenges posed by the correctness error which is inherent to all lattice-based constructions.

While our construction is mostly of theoretical interest, we believe it to be an important stepping stone for future works in this area.

Keywords: Blind Signatures, Forking Lemma, Lattices

1 Introduction

Blind Signatures. Blind signatures, first proposed by Chaum [19], are a fundamental cryptographic primitive with many applications such as eVoting [54], eCash [19], anonymous credentials [20,46,15,17,16,9,7], and, as of late, privacy preserving protocols in the context of blockchain protocols [61]. Informally, a blind signature scheme is an interactive protocol between a *signer* S (holding a secret key sk) and a *user* U (holding a public key pk and a message m) with the goal that U obtains a signature σ on m. The protocol should satisfy correctness (i.e., σ can be verified using the public key pk of S and m), unforgeability (i.e., only S can issue signatures), and blindness (i.e., S is not able to link σ to a particular execution of the protocol in which it was created). Blind signatures are among the most well-studied cryptographic primitives and it is well known how to construct blind signatures from general complexity assumptions [34,24,23]. However, achieving *efficient constructions* from *standard assumptions* is known to be a notoriously difficult task with only a handful of constructions being known. To make matters worse, even among these works, some have been pointed out to contain flawed security proofs [2,56]. Effectively, this leaves only the original works due to Pointcheval and Stern [51,50,52,53] based on Schnorr [57] and Okomoto-Schnorr [44] signatures.

Blind Signatures from Lattices. In this work, we revisit the problem of constructing blind signatures from standard *lattice assumptions*. This question was first addressed by Rückert [56], who gave a candidate construction based on Lyubashevsky's identification scheme [38] from the SIS assumption. Unfortunately, as we will explain in Section 1.2, his security proof contains a subtle flaw. While the recent work of Hauck, Kiltz, and Loss [33] introduces a general framework to obtain blind signatures from (collision resistant) linear hash functions, their framework does not cover the setting of lattice assumptions. Informally, the reason for this is that in the context of lattice-based constructions, most known cryptographic primitives exhibit some form of *noticeable correctness error*. Indeed, this is also true for Lyubashevsky's identification scheme/linear hash function implicitly used in [56]. This makes it impossible to apply the analysis of [33] directly, since it crucially relies on the fact that if both S and

U behave honestly, U always obtains a valid signature. Since [56] was published, more lattice-based constructions of blind signatures have been proposed. As we will discuss in detail below, all of these schemes either inherit the proof errors from [56] or introduce new ones. The main goal of our work is to give the first direct lattice-based blind signature scheme with a correct security proof.

1.1 Our Contributions

We construct a blind signature scheme from any linear hash functions [6,33] with noticeable correctness error. We use the aborting technique introduced by Alkadri, El Bansarkhani, and Buchmann [5] to reduce the correctness error of the blind signature scheme. Instantiating our construction with Lyubashevsky's linear hash function [38] we obtain a lattice-based blind signature scheme from the SIS assumption.

While our work offers the first correct proof for a lattice-based blind signature scheme, it comes with several severe drawbacks. First, we can only prove blindness in the weaker *honest signer model* [34] as compared to the *malicious signer model* [24]. We leave the construction of a scheme in the malicious signer setting as an open problem. Second, our construction comes with an exponential security loss in the reduction from the underlying hardness assumption (here, the SIS assumption). This is inherited from the proof technique of Pointcheval and Stern in the discrete logarithm setting [53]. This strongly restricts the number of signatures that can be issued per public key to a poly-logarithmic amount. Indeed, a sub-exponential attack due to Schnorr and Wagner [58,60] resulting from the ROS[4] problem shows that for the Schnorr and Okamoto-Schnorr blind signature schemes, these parameters are optimal. Extending [58,33], we are also able to relate the security of our blind signature to a Generalized ROS problem whose hardness is independent of the SIS problem. However, the sub-exponential attack of Schnorr and Wagner cannot be directly translated to the Generalized ROS problem due to the algebraic structure of our lattice-based instantiation (see Section 7). Therefore, an interesting open question is whether our "lattice" variant of the Generalized ROS problem can be solved in sub-exponential time. Nevertheless, we believe that our scheme makes an important first step toward future endeavors in this area by giving the first comprehensive and modular security proof for a blind signature scheme from lattice assumptions. While our scheme might not be practical by itself (our example instantiation has signatures sizes of roughly a couple of mega bytes),[5] it seems reasonable to apply similar ideas as in [49] to extend the number of allowed sessions per public key to a polynomial amount at not much overhead (but at the restriction of issuing signatures in a sequential fashion).

1.2 Problems with Existing Schemes

In the following we will first explain in detail the problems in the proof of Rückert's lattice-based blind signature scheme and then sketch how these errors propagate to subsequent schemes. We also list some other lattice-based constructions which have been found to be incorrect.

RÜCKERT'S BLIND SIGNATURE SCHEME. The key idea in the proof of Rückert's lattice-based blind signature scheme [56] is to rewind the forger (with partially different random oracles) so as to obtain two distinct values χ and χ' satisfying $\mathsf{F}(\chi) = \mathsf{F}(\chi')$, i.e., a collision in the underlying linear hash function. (In the lattice setting, a collision in the hash function directly implies a non-trivial solution $\chi - \chi'$ to the SIS problem.) To argue that $\chi \neq \chi'$, [56] attempts to apply the general forking lemma of Bellare and Neven [10] to the forger and argues that witness indistinguishability alone is sufficient to ensure $\chi \neq \chi'$. Here, [56] relies on Lemma 8 from Pointcheval and Stern's proof [53], who followed a similar approach. However, Lemma 8 does not state that χ and χ' are distinct; only that (by witness indistinguishability of their scheme) there exist two distinct secret keys sk, sk', which can lead to identical transcripts. This is *insufficient* to ensure $\chi \neq \chi'$ in the subsequent rewinding step. In fact, the Generalized ROS attack mentioned above works independently of the concrete secret key that is being used. Using this attack, it is always possible to force an outcome of $\chi = \chi'$ if the number of signatures per public key becomes larger than polylogarithmic in the security parameter. The crucial argument toward proving $\chi \neq \chi'$ only follows from Lemma 9 and the subsequent parts of Pointcheval and Stern's proof and is completely missing from Rückert's proof. It relies on a very subtle probabilistic method argument that only works in a (small) range of parameters for which the ROS problem remains information theoretically hard. Moreover, both Lemma 8 and 9 of [53] apply *exclusively* to the Okamoto-Schnorr scheme and cannot be transferred to other schemes directly. Adapting these lemmas to a setting with correctness error is one of the key novelties in our proof.

[4] ROS stands for Random inhomogenities in an Overdetermined, Solvable system of linear equations.

[5] It is not even clear how much better our scheme performs compared to generic constructions using non-interactive zero-knowledge proofs [24].

BLAZE AND BLAZE^+. BLAZE by Alkadri, El Bansarkhani, and Buchmann [4] improves Rückert's construction in the following two aspects. Firstly, BLAZE applies Gaussian rejection sampling [39] instead of uniform [38]. Secondly, it introduces the concept of signed permutations which allows to get rid of rejection sampling on the unblinded challenge values. While BLAZE introduces several interesting new concepts for constructing lattice-based blind signatures, its security analysis reuses the (incorrect) security arguments of Rückert at a crucial point in the reduction, and hence inherits its problems. Concretely, in the one-more unforgeability proof of [4, Theorem 3] it is missing the argument that the candidate solution for the inhomogeneous RSIS problem computed in Case 2 is non-trivial. Even worse and independent of the aforementioned problems with the proof, BLAZE is not one-more unforgeable as we will sketch now. Consider a user U interacting with the signer S in the one-more unforgeability experiment. At the end of the protocol execution, an honest U performs rejection sampling on some values $(\hat{z}_1, \hat{z}_2)$ contained in the signature. (Rejection sampling on U's side is used to ensure blindness.) If rejection sampling rejects, U sends the random coins used for rejection sampling as a proof to S which, upon successful verification, triggers a restart of the protocol. However, even in case rejection sampling rejects, the signature can still be valid and which case a dishonest U can trigger a restart of the protocol while still learning a valid signature. Since by the restart of the protocol U learns another valid signature, this observation can be turned into a simple one-more unforgeability attack. The aforementioned attack on BLAZE actually disappears in the recently proposed BLAZE^+ protocol [5] because U performs multiple rejection samplings in parallel and the probability that all of them reject becomes negligible. BLAZE^+ introduces a new technique of reducing correctness error by performing multiple rejection samplings in parallel in order to reduce the communication complexity. Unfortunately, the security analysis also reuses the (incorrect) security arguments of Rückert and hence inherits its problems.

FURTHER SCHEMES. Three recent works [14,37,47] propose new lattice-based blind signatures, but they also rely on the same analysis as Rückert to argue that a collision can be found with non-negligible probability when rewinding (see above). Unfortunately, this implies that all of these schemes do not have a valid security proof. There has been a line of research on lattice-based blind signatures using preimage sampleable trapdoor functions [21,63,62,30,29]. As shown by [4], all these schemes are insecure. Concretely, they give attacks which either recover the secret key or solve the underlying lattice problem in at most two executions of the signing protocol.

1.3 Related Work

Three round blind signatures are not achievable in the standard model [26]. We circumvent their result by using (programmable) random oracles and therefore believe that our proof strategies cannot be easily extended to the standard model. Blind signatures are impossible to construct from one-way permutations [35], even in the random oracle model. We circumvent their result by relying on the stronger assumption of collision resistance. A large class of Schnorr-type blind signature schemes cannot be proved secure if the underlying identification scheme has a unique witness [8]. We circumvent their result by requiring our underlying hard problem to have multiple witnesses corresponding to each public key.

As already mentioned, round optimal blind signatures can be constructed from general complexity assumptions beyond one-way permutations [34,24,32,23]. The impossibility results of [26] are circumvented by either relying on a CRS [24] or using complexity leveraging [32,23]. We refer to [23] for a detailed discussion on the topic of constructing blind signatures from general assumptions.

Several works [51,53,2,13,45,8,31,32,27] show how to construct efficient blind signatures schemes from concrete assumptions in the setting of prime-order groups, in some cases relying on bilinear maps.

1.4 Organization

After establishing some preliminaries in Section 2, in Section 3 we will introduce the notion of linear hash functions LHF with noticeable correctness error. In Section 4 we will define syntax and security of canonical (three-round) blind signature schemes. Figure 5 constructs a blind signature scheme $\mathsf{BS}_\eta[\mathsf{LHF}, \mathsf{H}, \mathsf{G}]$ from any linear hash function LHF and two standard hash functions H and G. This section also contains our main theorems about one-more unforgeability (Theorem 1) and blindness (Theorem 2). As a first step in the proof of Theorem 1, in Section 5 we will reduce the one-more unforgeability of $\mathsf{BS}_{\eta,\nu,\mu}[\mathsf{LHF}, \mathsf{H}, \mathsf{G}]$ to one-more man-in-the-middle security of the underlying canonical identification scheme $\mathsf{ID}_{\eta'}[\mathsf{LHF}]$ in the random oracle model. Appendix A contains the main technical part of this work, the proof of the one-more man-in-the-middle security of $\mathsf{ID}_{\eta'}[\mathsf{LHF}]$. In Section 6 we will provide an example instantiation of our framework based on the standard SIS assumption. Finally, Section 7 generalizes the ROS attack to our setting and proves that any attack on it also implies an attack on the one-more unforgeability of $\mathsf{BS}_{\eta,\nu,\mu}[\mathsf{LHF}, \mathsf{H}, \mathsf{G}]$.

2 Preliminaries and Notation

SETS AND VECTORS. For $n \in \mathbb{N}$, $[n]$ denotes the set $\{1, \ldots, n\}$. We use bold-faced, lower case letters $\boldsymbol{h}$ to denote a vector of elements and denote the length of $\boldsymbol{h}$ as $|\boldsymbol{h}|$. For $j \geq 1$, we write $\boldsymbol{h}_j$ to denote the j-th element of $\boldsymbol{h}$ and we write $\boldsymbol{h}_{[j]}$ to refer to the first j entries of $\boldsymbol{h}$, i.e., the elements $\boldsymbol{h}_1, ..., \boldsymbol{h}_j$. We use boldface, upper case letters $\mathbf{A}$ to denote matrices. We denote the i-th row of $\mathbf{A}$ as $\mathbf{A}_i$ and the j-th entry of $\mathbf{A}_i$ as $\mathbf{A}_{i,j}$. We let $\Delta(X, Y)$ indicate the statistical distance between two distributions X, Y.

SAMPLING FROM SETS. We write $h \xleftarrow{\$} \mathcal{S}$ to denote that the variable h is uniformly sampled from the finite set $\mathcal{S}$. For $1 \leq j \leq Q$ and $\boldsymbol{g} \in \mathcal{S}^{j-1}$, we write $\boldsymbol{h'} \xleftarrow{\$} \mathcal{S}^Q | \boldsymbol{g}$ to denote that the vector $\boldsymbol{h'}$ is uniformly sampled from $\mathcal{S}^Q$, conditioned on $\boldsymbol{h'}_{[j-1]} = \boldsymbol{g}$. This sampling process can be implemented by copying vector $\boldsymbol{g}$ into the first $j-1$ entries of $\boldsymbol{h}'$ and next sampling the remaining $Q - j + 1$ entries of $\boldsymbol{h}$, (i.e., $\boldsymbol{h}'_j, \ldots, \boldsymbol{h}'_Q \xleftarrow{\$} \mathcal{S}^{Q-j+1}$).

ALGORITHMS. We use uppercase, serif-free letters A, B to denote algorithms. Unless otherwise stated, algorithms are probabilistic and we write $(y_1, \ldots) \xleftarrow{\$} \mathsf{A}(x_1, \ldots)$ to denote that A returns $(y_1, \ldots)$ when run on input $(x_1, \ldots)$. We write A^{B} to denote that A has oracle access to B during its execution. To make the randomness ω of an algorithm A on input x explicit, we write $\mathsf{A}(x; \omega)$. Note that in this notation, A is deterministic. For a randomised algorithm A, we use the notation $y \in \mathsf{A}(x)$ to denote that y is a possible output of A on input x.

SECURITY GAMES. We use standard code-based security games [12]. A *game* $\mathbf{G}$ is a probability experiment in which an adversary A interacts with an implicit challenger that answers oracle queries issued by A. $\mathbf{G}$ has one *main procedure* and an arbitrary amount of additional *oracle procedures* which describe how these oracle queries are answered. To distinguish game-related oracle procedures from algorithmic procedures more clearly, we denote the former using monospaced font, e.g., `Oracle`. We denote the (binary) output b of game $\mathbf{G}$ between a challenger and an adversary A as $\mathbf{G}^{\mathsf{A}} \Rightarrow b$. A is said to *win* $\mathbf{G}$ if $\mathbf{G}^{\mathsf{A}} \Rightarrow 1$. Unless otherwise stated, the randomness in the probability term $\Pr[\mathbf{G}^{\mathsf{A}} \Rightarrow 1]$ is over all the random coins in game $\mathbf{G}$.

ALGEBRA. We let $\oplus$ denote the bitwise XOR operation. 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) $1_{\mathcal{S}} \cdot x = x$.

SECURITY NOTIONS. We formalize all security notions relative to some fixed parameters par. This streamlines the exposition considerably. In doing so, we consider a non-uniform notion of security, as the RSIS problem is not hard for fixed par, but only for par drawn (uniformly) at random in the security experiment. This is comparable to considerations as in [55]. However, we remark that using the splitting lemma our theorems can easily be made to work in a setting where par is indeed chosen at random along with the remaining (random) parts.

3 Linear Hash Functions

In this section we define *linear hash function families with correctness error* which are a generalization of linear (hash) function families with perfect correctness [6,33].

SYNTAX. A *linear hash function family* LHF 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

$$\mathcal{S} = \mathcal{S}(par), \quad \mathcal{D} = \mathcal{D}(par), \quad \text{and } \mathcal{R} = \mathcal{R}(par),$$

where $\mathcal{S}$ is a set of scalars such that $\mathcal{D}$ and $\mathcal{R}$ are modules over $\mathcal{S}$. The parameters par also define 9 *filter sets*

$$\mathcal{S}_{\mathsf{xxx}} \subseteq \mathcal{S} \ (\mathsf{xxx} \in \{\beta, c, c'\}) \text{ and } \mathcal{D}_{\mathsf{yyy}} \subseteq \mathcal{D} \ (\mathsf{yyy} \in \{sk, r, s, s', \alpha\}).$$

Throughout the paper, we will assume that par is fixed and implicitly given to all algorithms. For linear hash function families with perfect correctness [33], the filter sets are trivial, i.e., $\mathcal{S}_{\mathsf{xxx}} = \mathcal{S}$ and $\mathcal{D}_{\mathsf{yyy}} = \mathcal{D}$. Algorithm $\mathsf{F}(par, \cdot)$ 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 $x, y \in \mathcal{D}$ and $s \in \mathcal{S}$:

$$\mathsf{F}(s \cdot x + y) = s \cdot \mathsf{F}(x) + \mathsf{F}(y) \ . \tag{1}$$

We now define the technical conditions of *torsion-freeness*, *regularity*, *enclosedness*, and *smoothness* of LHF that will be useful for proving correctness and security of blind signatures constructed from LHF.

TORSION-FREENESS AND REGULARITY. We say that LHF 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 $c_1, c_2 \in \mathcal{S}_c$ satisfying $(c_1 - c_2) \cdot z^* = 0$ we have $c_1 - c_2 = 0$. Note that the existence of such an element implies that F is a many-to-one mapping.

We call LHF (ε, Q')-regular, if for all par generated with PGen, there exist sets $\mathcal{D}'_{sk}, \mathcal{D}'_r$ and a torsion-free element from the kernel z^* s.t.

$$\frac{|\mathcal{D}'_{sk}|}{|\mathcal{D}_{sk}|} \cdot \left(\frac{|\mathcal{D}'_r|}{|\mathcal{D}_r|}\right)^{Q'} \geq 1 - \varepsilon/4,$$

and where

$$\mathcal{D}'_{sk} := \{sk \in \mathcal{D}_{sk} : sk + z^* \in \mathcal{D}_{sk}\}$$

and

$$\mathcal{D}'_r := \{r \in \mathcal{D}_r : \forall c \in \mathcal{S}_c, r + cz^* \in \mathcal{D}_r\}.$$

Similar to the work of Hauck et al. [33], our proof of one-more unforgeability uses torsion-freeness and regularity to argue that a transcript of the scheme with a secret key sk can be preserved when switching to a different (valid) secret key $sk' := sk + z^*$, with high probability.

ENCLOSEDNESS ERROR. We say that LHF has *enclosedness errors* $(\delta_1, \delta_2, \delta_3)$ if for all $par \in \mathsf{PGen}(1^\kappa)$, $c' \in \mathcal{S}_{c'}, c \in \mathcal{S}_c, s \in \mathcal{D}_s, sk \in \mathcal{D}_{sk}$,

$$\Pr_{\beta \overset{\$}{\leftarrow} \mathcal{S}_\beta}[\beta + c' \notin \mathcal{S}_c] < \delta_1, \quad \Pr_{r \overset{\$}{\leftarrow} \mathcal{D}_r}[c \cdot sk + r \notin \mathcal{D}_s] < \delta_2, \text{ and } \Pr_{\alpha \overset{\$}{\leftarrow} \mathcal{D}_\alpha}[\alpha + s \notin \mathcal{D}_{s'}] < \delta_3.$$

The enclosedness error of LHF is directly linked to the *correctness error* of our schemes. Intuitively, the smaller this error, the easier it is to get a scheme which almost always works correctly.

SMOOTHNESS. We say that LHF is *smooth* if the following conditions hold for all $par \in \mathsf{PGen}(1^\kappa)$:

(S1) For all $s \in \mathcal{D}_s$ and $s' \in \mathcal{D}_{s'}$, we have $\|s' - s\|_\infty \in \mathcal{D}_\alpha$

(S2) For all $s_1, s_2 \in \mathcal{D}_s$ and random variables $\alpha^* \overset{\$}{\leftarrow} \{\alpha \in \mathcal{D}_\alpha \mid \alpha + s_1 \in \mathcal{D}_{s'}\}$, $\hat{\alpha} \overset{\$}{\leftarrow} \{\alpha \in \mathcal{D}_\alpha \mid \alpha + s_2 \in \mathcal{D}_{s'}\}$ we have that $\hat{\alpha} + s_2$ and $\alpha^* + s_1$ are identically distributed.

(S3) For all $s_1, s_2 \in \mathcal{D}_s$ and random variables $\underline{\alpha}^* \overset{\$}{\leftarrow} \{\alpha \in \mathcal{D}_\alpha \mid \alpha + s_1 \notin \mathcal{D}_{s'}\}$, $\underline{\hat{\alpha}} \overset{\$}{\leftarrow} \{\alpha \in \mathcal{D}_\alpha \mid \alpha + s_2 \notin \mathcal{D}_{s'}\}$ we have that $\underline{\hat{\alpha}} + s_2$ and $\underline{\alpha^*} + s_1$ are identically distributed.

(S4) For all $c' \in \mathcal{S}_{c'}$ and $c \in \mathcal{S}_c$, we have $\|c - c'\|_\infty \in \mathcal{S}_\beta$.

(S5) For all $c'_1, c'_2 \in \mathcal{S}_{c'}$ and random variables $\beta^* \overset{\$}{\leftarrow} \{\beta \in \mathcal{S}_\beta \mid \beta + c'_1 \in \mathcal{S}_c\}$, $\hat{\beta} \overset{\$}{\leftarrow} \{\beta \in \mathcal{S}_\beta \mid \beta + c'_2 \in \mathcal{S}_c\}$ we have that $\hat{\beta} + c'_2$ and $\beta^* + c'_1$ are identically distributed.

(S6) For all $c'_1, c'_2 \in \mathcal{S}_{c'}$ and random variables $\underline{\beta^*} \overset{\$}{\leftarrow} \{\beta \in \mathcal{S}_\beta \mid \beta + c'_1 \notin \mathcal{S}_c\}$, $\underline{\hat{\beta}} \overset{\$}{\leftarrow} \{\beta \in \mathcal{S}_\beta \mid \beta + c'_2 \notin \mathcal{S}_c\}$ we have that $\underline{\hat{\beta}} + c'_2$ and $\underline{\beta^*} + c'_1$ are identically distributed.

Smoothness of LHF will be a crucial tool for proving *blindness* of our schemes. Intuitively, smoothness allows to 'match' any message/signature pair (m_i, σ_i) that was generated via the i^{th} run of the scheme to the transcript T_j of *any run* $j \in \{1, ..., i, ...\}$.

COLLISION RESISTANCE. We say that LHF is (ε, t)-**CR** relative to $par \in \mathsf{PGen}(1^\kappa)$ if for all adversaries running in time at most t,

$$\Pr_{(x_1,x_2) \overset{\$}{\leftarrow} \mathsf{A}(par)}[(\mathsf{F}(x_1) = \mathsf{F}(x_2)) \wedge (x_1 \neq x_2) \wedge (x_1, x_2 \in \mathcal{D}')] \leq \varepsilon$$

where

$$\mathcal{D}' := \{s' - c' \cdot sk : s' \in \mathcal{D}_{s'}, c' \in \mathcal{S}_{c'}, sk \in \mathcal{D}_{sk}\} \subseteq \mathcal{D}. \tag{2}$$

4 Canonical Blind Signature Schemes

In this section, we recall syntax and security of a special type of blind signature scheme, called *canonical three-move blind signature scheme* [33]. In Section 4.1, we first recall the syntax of such schemes and give the proper security definitions. Next, in Section 4.3, we give a generic construction that gives a canonical three-move blind signature scheme $\mathsf{BS}[\mathsf{LHF}]$ from any linear hash function family LHF.

4.1 Definitions

Definition 1 (Canonical Three-Move Blind Signature Scheme). *A* canonical three-move blind signature scheme BS *is a tuple of algorithms* $\mathsf{BS} = (\mathsf{PGen}, \mathsf{KG}, \mathsf{S}, \mathsf{U}, \mathsf{BSVer})$.

- *The randomised* parameter generation algorithm PGen *returns system parameters par.*
- *The randomised* key generation algorithm KG *takes as input system parameters par and outputs a public key/secret key pair (pk, sk). We assume that pk defines a* challenge set $\mathcal{C} := \mathcal{C}(pk)$ *and that pk is known to all parties.*
- *The* signer algorithm S *is split into two algorithms, i.e.,* $\mathsf{S} := (\mathsf{S}_1, \mathsf{S}_2)$, *where:*
  - *The randomised algorithm S_1 takes as input the secret key sk and returns a commitment R and the signer's state stS.*
  - *The deterministic algorithm S_2 takes as input the signer's state stS, a secret key sk, a commitment R, and a challenge $c \in \mathcal{C}$. It returns the response s.*
- *The* user algorithm U *is split into two algorithms, i.e.,* $\mathsf{U} := (\mathsf{U}_1, \mathsf{U}_2)$, *where:*
  - *The randomised algorithm U_1 takes as input the public key pk, a commitment R, and a message m. It returns the user's state stU and a challenge $c \in \mathcal{C}$.*
  - *The deterministic algorithm U_2 takes as input the public key pk, a commitment R, a challenge $c \in \mathcal{C}$, a response s, a message m, and the user's state stU. It returns a signature σ where, possibly, $\sigma = \bot$.*
- *The deterministic verification algorithm* BSVer *takes as input the public key pk, a signature σ, and a message m. It outputs 1 (accept) or 0 (reject). We make the convention that* BSVer *always outputs 0 on input a signature $\sigma = \bot$.*

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

Consider an interaction $(R, c, s, \sigma) \leftarrow \langle \mathsf{S}(sk), \mathsf{U}(pk, m) \rangle$ between signer S and user U, as defined in Figure 1. We say that $\mathsf{BS} = (\mathsf{PGen}, \mathsf{KG}, \mathsf{S}, \mathsf{U}, \mathsf{BSVer})$ has *correctness error δ*, if for all messages $m \in \{0,1\}^*$, $par \in \mathsf{PGen}(1^\kappa)$, $(pk, sk) \in \mathsf{KG}(par)$,

$$\Pr_{(a,\sigma) \xleftarrow{\$} \langle \mathsf{S}(sk), \mathsf{U}(pk,m) \rangle} \left[\mathsf{BSVer}(pk, m, \sigma) \neq 1 \right] \leq \delta \, .$$

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

Fig. 1. Interaction $(R, c, s, \sigma) \leftarrow \langle \mathsf{S}(sk), \mathsf{U}(pk, m) \rangle$ between signer S and user U.

SECURITY NOTIONS. Security of a Canonical Three-Move Blind Signature Scheme BS is captured by two security notions: *blindness* and *one-more unforgeability*.

```
Game $\mathbf{Blind}_{\mathsf{BS},par}$:
$b \xleftarrow{\$} \{0,1\}$
$\boldsymbol{b}_1 \leftarrow b$; $\boldsymbol{b}_2 \leftarrow 1 - b$
$(sk, pk) \xleftarrow{\$} \mathsf{KG}(par)$
$b' \xleftarrow{\$} \mathsf{A}^{\mathtt{Init},\mathtt{U}_1,\mathtt{U}_2}(pk, sk)$
Return $b = b'$

Oracle $\mathtt{Init}(\tilde{m}_0, \tilde{m}_1)$: //Once
$\boldsymbol{m}_0 \leftarrow \tilde{m}_0, \boldsymbol{m}_1 \leftarrow \tilde{m}_1$
$\mathbf{sess}_1 \leftarrow \mathbf{sess}_2 \leftarrow \mathtt{init}$

Oracle $\mathtt{U}_1(sid, R)$:
If $sid \notin \{1,2\} \vee \mathbf{sess}_{sid} \neq \mathtt{init}$:
    Return $\bot$
$\mathbf{sess}_{sid} \leftarrow \mathtt{open}$
$\boldsymbol{R}_{sid} \leftarrow R$
$(\boldsymbol{c}_{sid}, \boldsymbol{st}_{sid}) \xleftarrow{\$} \mathsf{U}_1(pk, \boldsymbol{R}_{sid}, \boldsymbol{m}_{\boldsymbol{b}_{sid}})$
Return $(sid, \boldsymbol{c}_{sid})$

Oracle $\mathtt{U}_2(sid, s)$:
If $\mathbf{sess}_{sid} \neq \mathtt{open}$: Return $\bot$
$\mathbf{sess}_{sid} \leftarrow \mathtt{closed}$
$\boldsymbol{s}_{sid} \leftarrow s$
$\boldsymbol{\sigma}_{\boldsymbol{b}_{sid}} \xleftarrow{\$} \mathsf{U}_2(pk, \boldsymbol{R}_{sid}, \boldsymbol{c}_{sid}, \boldsymbol{s}_{sid}, \boldsymbol{st}_{sid})$
If $\mathbf{sess}_1 = \mathbf{sess}_2 = \mathtt{closed}$:
    If $\boldsymbol{\sigma}_0 = \bot \vee \boldsymbol{\sigma}_1 = \bot$:
        Return $(\bot, \bot)$
    Return $(\boldsymbol{\sigma}_0, \boldsymbol{\sigma}_1)$
Return $(sid, \mathtt{closed})$
```

Fig. 2. Games defining $\mathbf{Blind}_{\mathsf{BS},par}$ for a canonical three-move blind signature scheme BS, with the convention that adversary A makes exactly one query to $\mathtt{Init}$ at the beginning of its execution.

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. We remark that we consider for this work the weaker notion of blindness in the *honest signer model* [34] as compared to the *malicious signer model* [24]. The difference between these two models is that in the honest signer model, the adversary obtains the keys from the experiment, whereas in the malicious signer model, the adversary gets to choose its own keys. Also, our notion does not capture security of blind signatures under aborts, where S or U may stop the interactive signing protocol prematurely [18,59]. The work of [25] proposes generic transformation to achieve such a stronger notion. We formalize the notion of blindness (for a canonical three-move blind signature scheme BS and for parameters $par \in \mathsf{PGen}$) via game $\mathbf{Blind}_{\mathsf{BS},par}$ depicted in Figure 2. In $\mathbf{Blind}_{\mathsf{BS},par}$, the game takes the role of the user and A takes the role of the signer. First, the game selects a random bit b which determines the order of adversarially chosen messages in both transcripts. It then runs A on a freshly generated key pair (pk, sk). A is given access to the three oracles $\mathtt{Init}, \mathtt{U}_1$ and $\mathtt{U}_2$. By convention, A first has to query oracle $\mathtt{Init}$. Subsequently, A may open at most two sessions. For each of these two sessions, A obtains corresponding transcripts $\boldsymbol{T}_1 = (\boldsymbol{R}_1, \boldsymbol{c}_1, \boldsymbol{s}_1)$ and $\boldsymbol{T}_2 = (\boldsymbol{R}_2, \boldsymbol{c}_2, \boldsymbol{s}_2)$. The game uses $\boldsymbol{m}_b$ and $\boldsymbol{m}_{1-b}$ to generate the transcripts $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$, respectively. If A honestly completes both sessions with the game, it obtains signatures $\boldsymbol{\sigma}_b$ and $\boldsymbol{\sigma}_{1-b}$ on messages $\boldsymbol{m}_b$ and $\boldsymbol{m}_{1-b}$. Note that A obtains $\boldsymbol{\sigma}_b$ and $\boldsymbol{\sigma}_{1-b}$ by calling $\mathtt{U}_2$ twice. More precisely, the first call to $\mathtt{U}_2$ closes the first session and the second call closes the second session. Once both sessions are closed, the game checks if A acted honestly in both of them and if so, returns the signatures $(\boldsymbol{\sigma}_b, \boldsymbol{\sigma}_{1-b})$. If instead A has behaved dishonestly and, as a result, $\boldsymbol{\sigma}_b = \perp$ or $\boldsymbol{\sigma}_{1-b} = \perp$ at the time of closing the second session, $\mathtt{U}_2$ returs $(\perp, \perp)$. At the end of the experiment, A has to guess the bit b. We define the advantage of adversary A in $\mathbf{Blind}_{\mathsf{BS},par}$ as $\mathbf{Adv}^{\mathbf{Blind}}_{\mathsf{BS},par}(\mathsf{A}) := \left|\Pr[\mathbf{Blind}^{\mathsf{A}}_{\mathsf{BS},par} \Rightarrow 1] - \frac{1}{2}\right|$.

Definition 2 (Perfect Blindness). *Let* BS *be a canonical three-move blind signature scheme. We say that* BS *is* perfectly blind *relative to par* $\in \mathsf{PGen}(1^\kappa)$ *if for all adversaries* A, $\mathbf{Adv}^{\mathbf{Blind}}_{\mathsf{BS},par}(\mathsf{A}) = 0$.

OMUF OF BLIND SIGNATURE SCHEMES. Intuitively, one-more unforgeability ensures that a user U can not produce even a single signature more than it should be able to learn from its interactions with the signer S. Our notion does not cover the stronger notion of *honest-user unforgeability* but a generic transformation from [59] can be applied to achieve it. We formalize the notion of one-more unforgeability (for a canonical three-move blind signature scheme BS and for all parameter $par \in \mathsf{PGen}$) via game $\mathbf{OMUF}_{\mathsf{BS},par}$ as depicted in Figure 3. In $\mathbf{OMUF}_{\mathsf{BS},par}$, an adversary A in the role of U is run on input the public key of the signer S and subsequently interacts with oracles that imitate the behaviour of S. A call to $\mathtt{S}_1$ returns a new session identifier sid 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 (at most) Q_{S_2} transcripts $(\boldsymbol{R}_k, \boldsymbol{c}_k, \boldsymbol{s}_k)$, where the challenges $\boldsymbol{c}_k$ are chosen by A. (The remaining (at most) 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 closed $Q_{\mathsf{S}_2}(\mathsf{A}) \leq Q_{\mathsf{S}_2}$ signer sessions (from which it should be able to compute $Q_{\mathsf{S}_2}(\mathsf{A})$ signatures). We define the advantage of adversary A in $\mathbf{OMUF}_{\mathsf{BS},par}$ as $\mathbf{Adv}^{\mathbf{OMUF}}_{\mathsf{BS},par}(\mathsf{A}) := \Pr[\mathbf{OMUF}^{\mathsf{A}}_{\mathsf{BS},par} \Rightarrow 1]$ and denote its running time as $\mathbf{Time}^{\mathbf{OMUF}}_{\mathsf{BS},par}(\mathsf{A})$.

We remark that the definition of OMUF security is only meaningful for blind signature schemes with negligible correctness error: If the scheme has noticeable correctness error, then even an honest adversary would not be able to produce even ℓ valid signatures after having interacted with ℓ signing sessions. Thus an adversary may learn less than ℓ signatures, but still has to come up with $\ell + 1$ signatures. This results in a significant weakening of the definition.

Definition 3 (One-More Unforgeability). *Let* BS *be a canonical three-move blind signature scheme. We say that* BS *is* $(\varepsilon, t, Q_{\mathsf{S}_1}, Q_{\mathsf{S}_2})$-**OMUF** *relative to par* $\in \mathsf{PGen}$ *if for all adversaries* A *satisfying*

$$\mathbf{Time}^{\mathbf{OMUF}}_{\mathsf{BS},par}(\mathsf{A}) \leq t, \quad Q_{\mathsf{S}_1}(\mathsf{A}) \leq Q_{\mathsf{S}_1}, \quad Q_{\mathsf{S}_2}(\mathsf{A}) \leq Q_{\mathsf{S}_2}, \tag{3}$$

we have $\mathbf{Adv}^{\mathbf{OMUF}}_{\mathsf{BS},par}(\mathsf{A}) \leq \varepsilon$.

4.2 Hash Trees

In this section we define hash trees to build trees of commitments similarly as in [5]. The main advantage of this technique is that it significantly reduces the probability of abort in the signing protocol by performing a rejection sampling [38] multiple times and representing each trial as a leaf of the hash tree.

```
Game $\mathbf{OMUF}_{\mathsf{BS},par}$:
$(sk, pk) \xleftarrow{\$} \mathsf{KG}(par)$
$sid \leftarrow 0$                                                   //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 $\exists i \neq j : \boldsymbol{m}_i = \boldsymbol{m}_j$: Return 0              //all messages have to be distinct
If $\exists i \in [\ell(\mathsf{A})] : \mathsf{BSVer}(pk, \boldsymbol{m}_i, \boldsymbol{\sigma}_i) = 0$: Return 0   //All signatures have to be valid
$Q_{\mathsf{S}_1}(\mathsf{A}) \leftarrow \#\{k \mid \mathbf{sess}_k = \mathtt{open}\}$                  //#abandoned signer sessions
$Q_{\mathsf{S}_2}(\mathsf{A}) \leftarrow \#\{k \mid \mathbf{sess}_k = \mathtt{closed}\}$                //#closed signer sessions
If $\ell(\mathsf{A}) \geq Q_{\mathsf{S}_2}(\mathsf{A}) + 1$: Return 1
Return 0

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

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

Fig. 3. Game $\mathbf{OMUF}_{\mathsf{BS},par}$ with adversary A.

Let $\mathsf{G}\colon \{0,1\}^* \mapsto \{0,1\}^{2\lambda}$ be a hash function. A hash tree $\mathsf{HT}[\mathsf{G}]$ associated to G is the tuple of three deterministic algorithms $(\mathsf{HashTree}, \mathsf{BuildAuth}, \mathsf{RootCalc})$ from Figure 4. Algorithm $\mathsf{HashTree}$ takes as input a list of commitments $\boldsymbol{v}$ and returns a sequence of nodes tree spanning the tree and the root root of the tree; Algorithm $\mathsf{BuildAuth}$ takes as input a list of indices as well as a tree and outputs an authentication path auth; Algorithm $\mathsf{RootCalc}$ takes as input a node and an authentication path auth and returns the root root of a hash tree.

Note that for all nodes $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_\ell)$ and for all indices $m \in [\ell]$, we have $\mathsf{RootCalc}(\boldsymbol{v}_m, \mathsf{auth}) = \mathsf{root}$, where $(\mathsf{root}, \mathsf{tree}) \leftarrow \mathsf{HashTree}(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_\ell)$ and $\mathsf{auth} \leftarrow \mathsf{BuildAuth}(m, \mathsf{tree})$.

```
Algorithm $\mathsf{HashTree}(\boldsymbol{v})$
$\ell \leftarrow |\boldsymbol{v}|$ ; $\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{\ell-1} \leftarrow \boldsymbol{v}$
$h \leftarrow \lceil \log(\ell) \rceil$
For $j \in \{0, \ldots, \ell - 1\}$:
    $\boldsymbol{t}_{0,j} \leftarrow \mathsf{G}(\boldsymbol{v}_j)$
For $i \in \{1, \ldots, h\}$:
    For $j \in \{0, \ldots, 2^{h-i} - 1\}$:
        $\boldsymbol{t}_{i,j} \leftarrow \mathsf{G}(\boldsymbol{t}_{i-1,2j}, \boldsymbol{t}_{i-1,2j+1})$
$\mathsf{root} \leftarrow \boldsymbol{t}_{h,0}$
$\mathsf{tree} \leftarrow (\boldsymbol{t}_{1,0}, \ldots, \boldsymbol{t}_{h,2^{h-1}-1})$
Return $(\mathsf{root}, \mathsf{tree})$

Algorithm $\mathsf{BuildAuth}(m, \mathsf{tree})$
$(\boldsymbol{t}_{1,0}, \ldots, \boldsymbol{t}_{h,2^{h-i}-1}) \leftarrow \mathsf{tree}$
For $i \in \{0, \ldots, h - 1\}$:
    $s \leftarrow \lfloor m/2^i \rfloor$
    $b \leftarrow s \mod 2$
    If $b = 0$
        $\boldsymbol{a}_i \leftarrow \boldsymbol{t}_{s+1}$
    Else
        $\boldsymbol{a}_i \leftarrow \boldsymbol{t}_{s-1}$
$\mathsf{auth} \leftarrow (m, \boldsymbol{a}_0, \ldots, \boldsymbol{a}_{h-1})$
Return $\mathsf{auth}$

Algorithm $\mathsf{RootCalc}(v, \mathsf{auth})$
$(m, \boldsymbol{a}_0, \ldots, \boldsymbol{a}_{h-1}) \leftarrow \mathsf{auth}$
$\boldsymbol{b}_0 \leftarrow \mathsf{G}(v)$
For $i \in \{1, \ldots, h\}$:
    $s \leftarrow \lfloor m/2^{i-1} \rfloor$
    $b \leftarrow s \mod 2$
    If $b = 0$
        $\boldsymbol{b}_i \leftarrow \mathsf{G}(\boldsymbol{b}_{i-1}, \boldsymbol{a}_{i-1})$
    Else
        $\boldsymbol{b}_i \leftarrow \mathsf{G}(\boldsymbol{a}_{i-1}, \boldsymbol{b}_{i-1})$
Return $\mathsf{root} := \boldsymbol{b}_h$
```

Fig. 4. Description of the algorithms for $\mathsf{HT}[\mathsf{G}] = (\mathsf{HashTree}, \mathsf{BuildAuth}, \mathsf{RootCalc})$ associated to G.

4.3 Blind Signature Schemes from Linear Hash Function Families

Let LHF be a linear hash function family and $\mathsf{H}\colon \{0,1\}^* \to \mathcal{C}$, $\mathsf{G}\colon \{0,1\}^* \to \{0,1\}^{2\lambda}$ be hash functions where $\mathcal{C} = \mathcal{S}_{c'}$. Let $\eta, \nu, \mu \in \mathbb{N}$ be repetition parameters. In the following we define mappings which convert a tuple of integers to an unique larger integer and vice versa. We define $\mathsf{2Int}_{\eta,\nu,\mu} : [\eta] \times [\nu] \times [\mu] \to [\eta\nu\mu]$ as the mapping $(i, j, k) \mapsto i + \eta \cdot (j - 1) + \eta\nu \cdot (k - 1)$, such that $\mathsf{2Int}_{\eta,\nu,\mu}(1, 1, 1) = 1$ and $\mathsf{2Int}_{\eta,\nu,\mu}(\eta, \nu, \mu) = \eta\nu\mu$.

Figure 5 shows how to construct a canonical three-move blind signature scheme $\mathsf{BS}_{\eta,\nu,\mu}[\mathsf{LHF}, \mathsf{G}, \mathsf{H}]$, where the hashtree algorithms $\mathsf{HT}[\mathsf{G}] = (\mathsf{HashTree}, \mathsf{BuildAuth}, \mathsf{RootCalc})$ are defined in Figure 4.

We begin by proving correctness of $\mathsf{BS}_{\eta,\nu,\mu}[\mathsf{LHF}, \mathsf{G}, \mathsf{H}]$.

Lemma 1 (Correctness). *Let* LHF *be a linear hash function family, let* $\mathsf{G}\colon \{0,1\}^* \to \{0,1\}^{2\lambda}$ *and* $\mathsf{H}\colon \{0,1\}^* \to \mathcal{C}$ *be hash functions and* $\mathsf{HT}[\mathsf{G}]$ *be a hash tree and* $\mathsf{BS} := \mathsf{BS}_{\eta,\nu,\mu}[\mathsf{LHF}, \mathsf{G}, \mathsf{H}]$. *If* LHF *has* enclosedness errors $(\delta_1, \delta_2, \delta_3)$ *then* BS *has* correctness error $\delta_1^\mu + \delta_2^\eta + \delta_3^\nu$.

Proof. Consider an execution of BS defined in Figure 5. From the definition of *enclosedness errors* $(\delta_1, \delta_2, \delta_3)$ it follows directly that the probability that during the execution lines 13, 31 and 43 abort are δ_2^η, δ_1^μ and δ_3^ν, respectively.

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

Algorithm $\mathsf{S}_1(sk)$:
For $i \in [\eta]$: $\boldsymbol{r}_i \xleftarrow{\$} \mathcal{D}_r; \boldsymbol{R}_i \leftarrow \mathsf{F}(\boldsymbol{r}_i)$
$\boldsymbol{stS} \leftarrow (\boldsymbol{r}_1, \ldots, \boldsymbol{r}_\eta); \boldsymbol{R} \leftarrow (\boldsymbol{R}_1, \ldots, \boldsymbol{R}_\eta)$
Return $(\boldsymbol{stS}, \boldsymbol{R})$

Algorithm $\mathsf{S}_2(sk, \boldsymbol{R}, c, \boldsymbol{stS})$:
$(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_\eta) \leftarrow \boldsymbol{stS}$
If $c \notin \mathcal{S}_c$:
    Return $\perp$
For $i \in [\eta]$:
    $\boldsymbol{s}_i \leftarrow c \cdot sk + \boldsymbol{r}_i$
    If $\boldsymbol{s}_i \in \mathcal{D}_s$:
        Return $s \leftarrow \boldsymbol{s}_i$
Return $\perp$

Algorithm $\mathsf{BSVer}(pk, \sigma, m)$:
$(c', s', \mathsf{auth}) \leftarrow \sigma$
$R' \leftarrow \mathsf{F}(s') - c' \cdot pk$
$\mathsf{root} \leftarrow \mathsf{RootCalc}(R', \mathsf{auth})$
If $(c' = \mathsf{H}(\mathsf{root}, m)) \wedge (s' \in \mathcal{D}_{s'})$:
    Return 1
Return 0
```

```
Algorithm $\mathsf{U}_1(pk, \boldsymbol{R}, m)$:
$(\boldsymbol{R}_1, \ldots, \boldsymbol{R}_\eta) \leftarrow \boldsymbol{R}$
$\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_\nu \xleftarrow{\$} \mathcal{D}_\alpha; \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_\mu \xleftarrow{\$} \mathcal{S}_\beta$
$\gamma \xleftarrow{\$} \mathbb{Z}_\eta$
For $(i, j, k) \in [\eta] \times [\mu] \times [\nu]$:
    $\boldsymbol{R}'_{i \oplus \gamma, j, k} \leftarrow \boldsymbol{R}_i + \mathsf{F}(\boldsymbol{\alpha}_k) + \boldsymbol{\beta}_j \cdot pk$
$(\mathsf{root}, \mathsf{tree}) \leftarrow \mathsf{HashTree}(\boldsymbol{R}'_{1,1,1}, \ldots, \boldsymbol{R}'_{\eta,\mu,\nu})$
$c' \xleftarrow{\$} \mathsf{H}(\mathsf{root}, m)$
For $j \in [\mu]$:
    $\boldsymbol{c}_j \leftarrow c' + \boldsymbol{\beta}_j$
    If $\boldsymbol{c}_j \in \mathcal{S}_c$:
        Return $(c \leftarrow \boldsymbol{c}_j, \boldsymbol{stU} \leftarrow (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_\nu, c', j, \gamma, \mathsf{tree}))$
Return $\perp$

Algorithm $\mathsf{U}_2(pk, \boldsymbol{R}, c, s, m, \boldsymbol{stU})$:
$(\boldsymbol{R}_1, \ldots, \boldsymbol{R}_\eta) \leftarrow R$
$(\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_\nu, c', j, \gamma, \mathsf{tree}) \leftarrow \boldsymbol{stU}$
If $s \notin \mathcal{D}_s$:
    Return $\perp$
Find $i \in [\eta]$: $\mathsf{F}(s) = c \cdot pk + \boldsymbol{R}_i$
Return $\perp$ if $i$ does not exist
For $k \in [\nu]$:
    $\boldsymbol{s}'_k \leftarrow s + \boldsymbol{\alpha}_k$
    If $\boldsymbol{s}'_k \in \mathcal{D}_{s'}$:
        $\mathsf{auth} \leftarrow \mathsf{BuildAuth}(\mathsf{2Int}_{\eta,\nu,\mu}(i \oplus \gamma, j, k), \mathsf{tree})$
        Return $\sigma \leftarrow (c', s', \mathsf{auth})$
Return $\perp$
```

Fig. 5. Construction of the canonical three-move blind signature scheme $\mathsf{BS} := \mathsf{BS}_{\eta,\nu,\mu}[\mathsf{LHF}, \mathsf{G}, \mathsf{H}]$ from a linear hash function family $\mathsf{LHF} = (\mathsf{PGen}, \mathsf{F})$, where $\mathsf{BS} := (\mathsf{PGen}, \mathsf{KG}, \mathsf{S} = (\mathsf{S}_1, \mathsf{S}_2), \mathsf{U} = (\mathsf{U}_1, \mathsf{U}_2), \mathsf{BSVer})$ and challenge set $\mathcal{C} := \mathcal{S}_{c'}$.

We continue with a statement about OMUF security of $\mathsf{BS}_{\eta,\nu,\mu}[\mathsf{LHF}, \mathsf{G}, \mathsf{H}]$. Its proof will be given in Section 5.

Theorem 1 (OMUF). *Let* $\mathsf{LHF} = (\mathsf{PGen}, \mathsf{F})$ *be a* $(\varepsilon, \eta\nu\mu Q_{\mathsf{S}_1})$*-regular linear hash function family with a torsion-free element from the kernel, let* $\mathsf{G}\colon \{0,1\}^* \to \{0,1\}^{2\lambda}$ *and* $\mathsf{H}\colon \{0,1\}^* \to \mathcal{C}$ *be random oracles. If* LHF *is* (ε', t')-**CR** *relative to* $par \in \mathsf{PGen}(1^\kappa)$*, then* $\mathsf{BS}_{\eta,\nu,\mu}[\mathsf{LHF}, \mathsf{G}, \mathsf{H}]$ *is* $(\varepsilon, t, Q_{\mathsf{S}_1}, Q_{\mathsf{S}_2}, Q_{\mathsf{G}}, Q_{\mathsf{H}})$-**OMUF** *relative to par in the random oracle model, where*

$$t' = 2t, \quad \varepsilon' = O\left(\left(\varepsilon^2 - \frac{Q_{\mathsf{G}}^2 + Q_{\mathsf{G}}}{2^\lambda} - \frac{(Q_{\mathsf{V}} Q_{\mathsf{P}_1})^{Q_{\mathsf{P}_2}+1}}{\left(\frac{|\mathcal{C}|}{|\mathcal{S}_c|}\right)^{Q_{\mathsf{P}_2}} \cdot |\mathcal{C}|}\right)^2 \frac{1}{Q_{\mathsf{V}}^2 Q_{\mathsf{P}_2}^3}\right),$$

Q_{G} *and* Q_{H} *are the number of queries to random oracles* G *and* H.

Theorem 2 (Blindness). *Let* $\mathsf{LHF} = (\mathsf{PGen}, \mathsf{F})$ *be a smooth linear hash function family and let* $\mathsf{G}\colon \{0,1\}^* \to \{0,1\}^{2\lambda}$ *and* $\mathsf{H}\colon \{0,1\}^* \to \mathcal{C}$ *be random oracles. Then* $\mathsf{BS}_{\eta,\nu,\mu}[\mathsf{LHF}, \mathsf{G}, \mathsf{H}]$ *is perfectly blind relative to all* $par \in \mathsf{PGen}(1^\kappa)$.

Let $\mathsf{BS} := \mathsf{BS}_{\eta,\nu,\mu}[\mathsf{LHF}, \mathsf{G}, \mathsf{H}]$. Intuitively the goal of an adversary in the $\mathbf{Blind}_{\mathsf{BS},par}$ experiment is as follows. The adversary interacts twice with the experiment and thus creates two transcripts. At the end of the interaction the adversary learns two message/signature pairs and tries to unblind which message/signature pair was created in which session. Intuitively to prevent the adversary from doing so, any combination of a transcript and a message/signature pair can be explained by some randomness (of the user) which (i) could have been used to create both the transcript and the message/signature pair and (ii) is indistinguishable from uniformly drawn randomness.

Proof. Fix two messages $\boldsymbol{m}_0, \boldsymbol{m}_1$ and let A be an adversary in the $\mathbf{Blind}_{\mathsf{BS},par}$ experiment (cf. Figure 2).

Given the output of an interaction $(\boldsymbol{R}_1, \ldots, \boldsymbol{R}_\eta, c, s, m, \sigma) \xleftarrow{\$} \langle \mathsf{S}(sk), \mathsf{U}(pk) \rangle$ we define a transcript $T := (\boldsymbol{R}_1, \ldots, \boldsymbol{R}_\eta, c, s)$. Consider A's view in an execution of $\mathbf{Blind}_{\mathsf{BS},par}$, which consists of the two transcripts $(\boldsymbol{T}_1, \boldsymbol{T}_2)$ and the two signatures $(\boldsymbol{\sigma}_0, \boldsymbol{\sigma}_1)$, where signature $\boldsymbol{\sigma}_b$ corresponds to transcript $\boldsymbol{T}_1$, signature $\boldsymbol{\sigma}_{1-b}$ corresponds to transcript $\boldsymbol{T}_2$, and b is the secret choice bit. Note that it is w.l.o.g. that $\boldsymbol{\sigma}_0, \boldsymbol{\sigma}_1 \neq \perp$. Now, the theorem is implied by the following two claims.

(B1) For each of the four combinations $(\boldsymbol{T}_{sid}, \boldsymbol{\sigma}_i)$, where $(sid, i) \in \{1,2\} \times \{0,1\}$, there exists randomness $\boldsymbol{rndU}_{sid,i} := (\boldsymbol{\alpha}_{sid,i,1} \ldots, \boldsymbol{\alpha}_{sid,i,\nu}, \boldsymbol{\beta}_{sid,i,1}, \ldots, \boldsymbol{\beta}_{sid,i,\mu}, \boldsymbol{\gamma}_{sid,i})$ of the user algorithm which results in the tuple $(\boldsymbol{T}_{sid}, \boldsymbol{\sigma}_i)$.

(B2) The real randomness $(\boldsymbol{rndU}_{1,b}, \boldsymbol{rndU}_{2,1-b})$ used in $\mathbf{Blind}_{\mathsf{BS},par}$ is identically distributed to the "fake" randomness $(\boldsymbol{rndU}_{1,1-b}, \boldsymbol{rndU}_{2,b})$.

To prove condition (B1) we argue as follows. Let $\mathsf{2Int}^{-1} : [\eta\nu\mu] \to [\eta] \times [\nu] \times [\mu]$ be the inverse of $\mathsf{2Int}_{\eta,\nu,\mu}$, defined in Section 4.3. Let $(\boldsymbol{c}'_i, \boldsymbol{s}'_i, \mathbf{auth}_i) \leftarrow \boldsymbol{\sigma}_i$. Let $(\boldsymbol{i}_i, \boldsymbol{j}_i, \boldsymbol{k}_i) \leftarrow \mathsf{2Int}^{-1}(\boldsymbol{n}_i)$, where $(\boldsymbol{n}_i, \boldsymbol{a}_{i,1}, \ldots, \boldsymbol{a}_{i,h}) \leftarrow \mathsf{auth}_i$. Define $\boldsymbol{\alpha}_{sid,i,\boldsymbol{k}_i} := \boldsymbol{s}'_{\boldsymbol{k}_i} - \boldsymbol{s}_{sid}$, $\boldsymbol{\beta}_{sid,i,\boldsymbol{j}_i} := \boldsymbol{c}_{sid} - \boldsymbol{c}'_{\boldsymbol{j}_i}$ and for all $\ell \in [\nu] \setminus \{\boldsymbol{k}_i\}$, $\boldsymbol{\alpha}_{sid,i,\ell} \xleftarrow{\$} \{\alpha \in \mathcal{D}_\alpha \mid \alpha + \boldsymbol{s}_{sid} \notin \mathcal{D}_{s'}\}$ for all $\ell \in [\mu] \setminus \{\boldsymbol{j}_i\}$, $\boldsymbol{\beta}_{sid,i,\ell} \xleftarrow{\$} \{\beta \in \mathcal{S}_\beta \mid \beta + \boldsymbol{c}_{sid} \notin \mathcal{S}_c\}$. Set $\boldsymbol{i}_{sid} \in [\eta]$ to be the smallest value s.t. $\mathsf{F}(\boldsymbol{s}_{sid}) = \boldsymbol{c}_{sid} \cdot pk + \boldsymbol{R}_{sid,\boldsymbol{i}_{sid}}$. Define $\boldsymbol{\gamma}_{sid,i} \leftarrow \boldsymbol{i}_{sid} \oplus \boldsymbol{i}_i$. By smoothness conditions (S1) and (S4) it follows that $\boldsymbol{\alpha}_{sid,i,\boldsymbol{k}_i} \in \mathcal{D}_\alpha$ and $\boldsymbol{\beta}_{sid,i,\boldsymbol{j}_i} \in \mathcal{S}_\beta$. Clearly, for all $\ell \in [\nu] \setminus \{\boldsymbol{k}_i\}$, $\boldsymbol{\alpha}_{sid,i,\ell} \in \mathcal{D}_\alpha$ for all $\ell \in [\mu] \setminus \{\boldsymbol{j}_i\}$, $\boldsymbol{\beta}_{sid,i,\ell} \in \mathcal{S}_\beta$. Clearly, $\boldsymbol{\gamma}_{sid,i} \in [\eta]$. Let $\boldsymbol{R}'_{sid,i,\boldsymbol{i}_i,\boldsymbol{j}_i,\boldsymbol{k}_i} = \boldsymbol{R}_{sid,\boldsymbol{i}_i} + \boldsymbol{\beta}_{sid,i,\boldsymbol{j}_j} \cdot pk + \mathsf{F}(\boldsymbol{\alpha}_{sid,i,\boldsymbol{k}_i})$. Let $\mathbf{root}_{sid,i} \leftarrow \mathsf{RootCalc}(\boldsymbol{R}'_{sid,i,\boldsymbol{i}_i,\boldsymbol{j}_i,\boldsymbol{k}_i}, \mathsf{auth}_i)$. To show that $\boldsymbol{c}'_i = \mathsf{H}(\mathbf{root}_{sid,i}, \boldsymbol{m}_i)$ we continue as follows. Since $\boldsymbol{T}_{sid}$ is a valid transcript, we have $\mathsf{F}(\boldsymbol{s}_{sid}) = \boldsymbol{R}_{sid,\boldsymbol{i}_i} + \boldsymbol{c}_{sid} \cdot pk$. Therefore,

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

Since $\boldsymbol{\sigma}_i$ is a valid signature we have $\boldsymbol{c}'_i = \mathsf{H}(\mathsf{RootCalc}(\mathsf{F}(\boldsymbol{s}'_i) - \boldsymbol{c}'_i \cdot pk, \mathbf{auth}_i), \boldsymbol{m}_i) = \mathsf{H}(\mathbf{root}_{sid,i}, \boldsymbol{m}_i)$.

To show condition (B2) we continue as follows. By smoothness condition (S2) if follows that $\boldsymbol{\alpha}_{1,b,\boldsymbol{k}_b}$ and $\boldsymbol{\alpha}_{2,1-b,\boldsymbol{k}_{1-b}}$ have the same distribution as $\boldsymbol{\alpha}_{1,1-b,\boldsymbol{k}_{1-b}}$ and $\boldsymbol{\alpha}_{2,b,\boldsymbol{k}_b}$. By smoothness condition (S5) it follows that $\boldsymbol{\beta}_{1,b,\boldsymbol{j}_b}$ and $\boldsymbol{\beta}_{2,1-b,\boldsymbol{j}_{1-b}}$ have the same distribution as $\boldsymbol{\beta}_{1,1-b,\boldsymbol{j}_{1-b}}$ and $\boldsymbol{\beta}_{2,b,\boldsymbol{j}_b}$. By smoothness condition (S3) for all $\ell \in [\nu] \setminus \{\boldsymbol{k}_b, \boldsymbol{k}_{1-b}\}$, $\boldsymbol{\alpha}_{1,b,\ell}$ and $\boldsymbol{\alpha}_{1,1-b,\ell}$ have the same distribution as $\boldsymbol{\alpha}_{2,b,\ell}$ and $\boldsymbol{\alpha}_{2,1-b,\ell}$. By smoothness condition (S6) for all $\ell \in [\mu] \setminus \{\boldsymbol{j}_b, \boldsymbol{j}_{1-b}\}$, $\boldsymbol{\beta}_{1,b,\ell}$ and $\boldsymbol{\beta}_{1,1-b,\ell}$ have the same distribution as $\boldsymbol{\beta}_{2,b,\ell}$ and $\boldsymbol{\beta}_{2,1-b,\ell}$. Clearly, all four $\gamma_{1,0}$, $\gamma_{1,1}$, $\gamma_{2,0}$ and $\gamma_{2,1}$ have the same distribution.

5 Proof of One-More Unforgeability

In this section, we will make a first step to the proof of Theorem 1, the one-more unforgeability of $\mathsf{BS}_{\eta,\nu,\mu}[\mathsf{LHF}, \mathsf{H}, \mathsf{G}]$ defined in Figure 5. To this end we first define canonical identification schemes and prove in Theorem 3 that one-more unforgeability of $\mathsf{BS}_{\eta,\nu,\mu}[\mathsf{LHF}, \mathsf{H}, \mathsf{G}]$ is implied by one-more man-in-the-middle security of the underlying identification scheme $\mathsf{ID}_{\eta'}[\mathsf{LHF}]$. Next, in Theorem 4 we will state that collision-resistance of LHF implies one-more man-in-the-middle security of the canonical identification scheme $\mathsf{ID}_{\eta'}[\mathsf{LHF}]$.

5.1 Canonical Identification Schemes

We recall syntax of *canonical (three-move) identification schemes* [1].

Definition 4 (Canonical Three-Move Identification Scheme). *A* canonical three-move identification scheme *is a tuple of algorithms* $\mathsf{ID} = (\mathsf{PGen}, \mathsf{KG}, \mathsf{P} = (\mathsf{P}_1, \mathsf{P}_2), \mathsf{IDVer})$.

- *The randomised* parameter generation algorithm PGen *returns system parameters* par.
- *The randomised* key generation algorithm KG *takes as input system parameters* par *and returns a public/secret key pair* (pk, sk). *We assume that* pk *implicitly defines a* challenge space $\mathcal{C} := \mathcal{C}(pk)$ *and that* pk *is distributed (and hence known) to all parties.*
- *The* prover algorithm P *is split into two algorithms, i.e.,* $\mathsf{P} := (\mathsf{P}_1, \mathsf{P}_2)$*, where:*
  - *The randomised algorithm* 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 secret key* sk*, a commitment* R*, a challenge* c*, and a state* st*. It returns a response* s.

– *The deterministic* verification algorithm IDVer *takes as input a public key pk, a commitment R, a challenge c, and a response s. It returns* 1 *(accept) or* 0 *(reject).*

Figure 6 shows the interaction between algorithms $\mathsf{P}_1, \mathsf{P}_2$, and IDVer. Since we will use ID only for the purpose of simplifying our main security statement, we refrain from giving the standard correctness definition.

Prover: sk		Verifier: pk
$(R, stP) \xleftarrow{\$} \mathsf{P}_1(sk)$	$\xrightarrow{R}$	
	$\xleftarrow{c}$	$c \xleftarrow{\$} \mathcal{C}$
$s \leftarrow \mathsf{P}_2(sk, R, c, stP)$	$\xrightarrow{s}$	$b \leftarrow \mathsf{IDVer}(pk, R, c, s)$
		Output b

Fig. 6. Interaction $(R, c, s) \leftarrow \langle \mathsf{P}(sk), \mathsf{IDVer}(pk) \rangle$ of a canonical three-move identification scheme $\mathsf{ID} = (\mathsf{PGen}, \mathsf{KG}, \mathsf{P}_1, \mathsf{P}_2, \mathsf{IDVer})$.

We now recall *One-More Man-in-the-Middle* security for canonical identification schemes [33]. The One-More Man-in-the-Middle ($\mathbf{OMMIM}$) security experiment for an identification scheme ID and an adversary A is defined in Figure 7. 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 $\mathtt{open}$. A call to $\mathtt{P}_2(pSid, \cdot)$ with the same $pSid$ sets the flag $\mathbf{pSess}_{pSid}$ to $\mathtt{closed}$. Similarly, a call to $\mathtt{V}_1$ returns a new verifier session identifier $vSid$ and sets flag $\mathbf{vSess}_{vSid}$ to $\mathtt{open}$. A call to $\mathtt{V}_2(vSid, \cdot)$ with the same $vSid$ sets the flag $\mathbf{vSess}_{vSid}$ to $\mathtt{closed}$. A closed verifier session $vSid$ is successful if the oracle $\mathtt{V}_2(vSid, \cdot)$ returns 1. Lines 04-07 define several internal random variables for later reference. Variable $Q_{\mathsf{P}_2}(\mathsf{A})$ 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 $\mathbf{OMMIM}_{\mathsf{ID},par}$ 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. A's advantage in $\mathbf{OMMIM}_{\mathsf{ID},par}$ is defined as $\mathbf{Adv}^{\mathbf{OMMIM}}_{\mathsf{ID},par}(\mathsf{A}) := \Pr[\mathbf{OMMIM}^{\mathsf{A}}_{\mathsf{ID},par} \Rightarrow 1]$ and we denote its running time as $\mathbf{Time}^{\mathbf{OMMIM}}_{\mathsf{ID},par}(\mathsf{A})$.

Definition 5 (One-more man-in-the-middle security). *We say that* ID *is* $(\varepsilon, t, Q_{\mathsf{V}}, Q_{\mathsf{P}_1}, Q_{\mathsf{P}_2})$-$\mathbf{OMMIM}$ *relative to* $par \in \mathsf{PGen}(1^\kappa)$ *if for all adversaries* A *satisfying* $\mathbf{Time}^{\mathbf{OMMIM}}_{\mathsf{ID},par}(\mathsf{A}) \leq t$, $Q_{\mathsf{V}}(\mathsf{A}) \leq Q_{\mathsf{V}}$, $Q_{\mathsf{P}_2}(\mathsf{A}) \leq Q_{\mathsf{P}_2}$, *and* $Q_{\mathsf{P}_1}(\mathsf{A}) \leq Q_{\mathsf{P}_1}$, *we have* $\mathbf{Adv}^{\mathbf{OMMIM}}_{\mathsf{ID},par}(\mathsf{A}) \leq \varepsilon$.

We remark that *security against impersonation under active and passive attacks* [1] is a weaker notion than OMMIM security, whereas *man-in-the-middle security* [11] is stronger.

5.2 Identification Schemes from Linear Hash Function Families

Let LHF be a linear hash function family and η' be a repetition parameter. Consider the canonical three-move blind signature scheme $\mathsf{BS}_{\eta,\nu,\mu}[\mathsf{LHF}, \mathsf{H}, \mathsf{G}] = (\mathsf{PGen}, \mathsf{KG}, \mathsf{S} = (\mathsf{S}_1, \mathsf{S}_2), \mathsf{U} = (\mathsf{U}_1, \mathsf{U}_2), \mathsf{BSVer})$ from Figure 5. BS directly implies a canonical identification scheme $\mathsf{ID}_{\eta'}[\mathsf{LHF}] = (\mathsf{PGen}, \mathsf{KG}, \mathsf{P}, \mathsf{IDVer})$ with challenge set $\mathcal{C} := \mathcal{S}_{c'}$, where prover P plays the role of the signer S, i.e., $\mathsf{P} = (\mathsf{P}_1, \mathsf{P}_2) := (\mathsf{S}_1, \mathsf{S}_2)$ and algorithm IDVer is defined as follows.

```
Algorithm IDVer($pk, \boldsymbol{R}_1, \ldots, \boldsymbol{R}_{\eta'}, c, s$):
For $i \in [\eta']$:
    If $(\boldsymbol{R}_i = \mathsf{F}(s) - c \cdot pk) \wedge (s \in \mathcal{D}_s)$:
        Return 1
Return 0
```

The identification scheme $\mathsf{ID}_{\eta'}[\mathsf{LHF}]$ can be seen as the projection of $\mathsf{BS}_{\eta,\nu,\mu}[\mathsf{LHF}, \mathsf{H}, \mathsf{G}]$ to the signer, i.e., all user algorithms (involving the techniques to achieve blindness) are removed. This makes it conceptually much simpler.

We will now show that OMUF security of $\mathsf{BS}_{\eta,\nu,\mu}[\mathsf{LHF}, \mathsf{G}, \mathsf{H}]$ is implied (in the ROM) by OMMIM security of $\mathsf{ID}_{\eta'}[\mathsf{LHF}]$, where $\eta' = \eta\nu\mu$.

```
Game $\mathbf{OMMIM}^{\mathsf{A}}_{\mathsf{ID},par}$:
01 $(sk, pk) \leftarrow \mathsf{KG}(par)$
02 $pSid \leftarrow 0, vSid \leftarrow 0$
03 $\mathsf{A}^{\mathtt{P}_1,\mathtt{P}_2,\mathtt{V}_1,\mathtt{V}_2}(pk)$
04 $Q_{\mathtt{Ch}}(\mathsf{A}) \leftarrow vSid$                                                         //#total sessions with verifier
05 $Q_{\mathsf{P}_1}(\mathsf{A}) \leftarrow \#\{1 \le k \le pSid \mid \mathbf{pSess}_k = \mathtt{open}\}$        //#abandoned prover sessions
06 $Q_{\mathsf{P}_2}(\mathsf{A}) \leftarrow \#\{1 \le k \le pSid \mid \mathbf{pSess}_k = \mathtt{closed}\}$      //#closed prover sessions
07 $\ell(\mathsf{A}) \leftarrow \#\{1 \le k \le vSid \mid \mathbf{vSess}_k = \mathtt{closed} \wedge \boldsymbol{b}'_k = 1\}$   //#successful verifier sessions
08 If $\ell(\mathsf{A}) \ge Q_{\mathsf{P}_2}(\mathsf{A}) + 1$: Return 1                                    //A's winning condition
09 Return 0

Oracle $\mathtt{P}_1$:
10 $pSid \leftarrow pSid + 1$
11 $\mathbf{pSess}_{pSid} \leftarrow \mathtt{open}$
12 $(\boldsymbol{R}_{pSid}, \boldsymbol{st}_{pSid}) \xleftarrow{\$} \mathsf{P}_1(sk)$
13 Return $(pSid, \boldsymbol{R}_{pSid})$

Oracle $\mathtt{P}_2(pSid, c)$:
14 If $\mathbf{pSess}_{pSid} \ne \mathtt{open}$: Return $\perp$
15 $\mathbf{pSess}_{pSid} \leftarrow \mathtt{closed}$
16 $s \leftarrow \mathsf{P}_2(sk, \boldsymbol{R}_{pSid}, c, \boldsymbol{st}_{pSid})$
17 Return $s$

Oracle $\mathtt{V}_1(R')$:
18 $vSid \leftarrow vSid + 1$
19 $\mathbf{vSess}_{vSid} \leftarrow \mathtt{open}$
20 $\boldsymbol{R}'_{vSid} \leftarrow R'; \boldsymbol{c}'_{vSid} \xleftarrow{\$} \mathcal{C}$
21 Return $(vSid, \boldsymbol{c}'_{vSid})$

Oracle $\mathtt{V}_2(vSid, s')$:
22 If $\mathbf{vSess}_{vSid} \ne \mathtt{open}$: Return $\perp$
23 $\mathbf{vSess}_{vSid} \leftarrow \mathtt{closed}$
24 $\boldsymbol{b}'_{vSid} \leftarrow \mathsf{IDVer}(pk, \boldsymbol{R}'_{vSid}, \boldsymbol{c}'_{vSid}, s')$
25 Return $\boldsymbol{b}'_{vSid}$
```

Fig. 7. The One-More Man-in-the-Middle security game $\mathbf{OMMIM}^{\mathsf{A}}_{\mathsf{ID},par}$

Theorem 3. *Let* LHF *be a linear hash function family, let* $\mathsf{G}\colon \{0,1\}^* \to \{0,1\}^{2\lambda}$ *and* $\mathsf{H}\colon \{0,1\}^* \to \mathcal{C}$ *be random oracles and let* $\mathsf{ID} := \mathsf{ID}_{\eta'}[\mathsf{LHF}]$, $\mathsf{BS} := \mathsf{BS}_{\eta,\nu,\mu}[\mathsf{LHF},\mathsf{G},\mathsf{H}]$. *If* ID *is* $(\varepsilon', t', Q_{\mathsf{V}}, Q_{\mathsf{P}_1}, Q_{\mathsf{P}_2})$-**OMMIM** *relative to* $par \in \mathsf{PGen}(1^\kappa)$ *then* BS *is* $(\varepsilon, t, Q_{\mathsf{S}_1}, Q_{\mathsf{S}_2}, Q_{\mathsf{G}}, Q_{\mathsf{H}})$-**OMUF** *relative to par in the random oracle model, where*

$$t' \approx t, \quad \varepsilon' = \varepsilon - \frac{Q_{\mathsf{G}}^2}{2^{2\lambda}} - \frac{Q_{\mathsf{G}}}{2^{2\lambda}}, \quad \eta' = \eta\nu\mu, \quad Q_{\mathsf{V}} = Q_{\mathsf{H}}, \quad Q_{\mathsf{P}_1} = Q_{\mathsf{S}_1}, \quad Q_{\mathsf{P}_2} = Q_{\mathsf{S}_2},$$

Q_{G} *and* Q_{H} *are the number of queries to random oracles* G *and* H*;*

Proof. Let A be an adversary that runs in the $\mathbf{OMUF}_{\mathsf{BS},par}$ experiment and breaks $(\varepsilon, t, Q_{\mathsf{S}_1}, Q_{\mathsf{S}_2}, Q_{\mathsf{G}}, Q_{\mathsf{H}})$-one-more-unforgeability of BS in the random oracle model. Consider a modified $\mathbf{OMUF}'_{\mathsf{BS},par}$ experiment in which random oracle G is replaced by the random oracle G depicted in Figure 8 which excludes collisions (i.e., the existence of queries $e \ne e'$ with $\mathsf{G}(e) = \mathsf{G}(e')$) and chains (i.e., the hash query $\mathsf{G}(\mathsf{G}(e))$ was made before the query $\mathsf{G}(e)$). Note that the statistical difference between games $\mathbf{OMUF}_{\mathsf{BS},par}$ and $\mathbf{OMUF}'_{\mathsf{BS},par}$ is bounded by $\frac{Q_{\mathsf{G}}^2}{2^{2\lambda}} + \frac{Q_{\mathsf{G}}}{2^{2\lambda}}$ and the number of queries to the oracles remains the same.

In Figure 8 we construct an adversary B that runs in the $\mathbf{OMMIM}_{\mathsf{ID},par}$ experiment and perfectly simulates A's oracles $\mathtt{S}_1$, $\mathtt{S}_2$, G, H via its own oracles $\mathtt{P}_1$, $\mathtt{P}_2$, and $\mathtt{Ch}$, respectively. Note that B calls $\mathtt{P}_2$ at most $Q_{\mathsf{P}_2} = Q_{\mathsf{S}_2}$ many times over the course of its simulation and moreover, $Q_{\mathsf{P}_2}(\mathsf{B}) = Q_{\mathsf{S}_2}(\mathsf{A})$. We show that B breaks $(\varepsilon', t', Q_{\mathsf{V}}, Q_{\mathsf{P}_1}, Q_{\mathsf{P}_2})$-OMMIM security of ID. Suppose that A is successful, i.e., it outputs $\ell(\mathsf{A}) \ge Q_{\mathsf{S}_2}(\mathsf{A}) + 1 = Q_{\mathsf{P}_2}(\mathsf{B}) + 1$ valid signatures on distinct messages and the number of closed sessions with the signer is at most $Q_{\mathsf{S}_2}(\mathsf{A}) = Q_{\mathsf{P}_2}(\mathsf{B})$.

Recursive algorithm $\mathsf{PreimageLeafs}_{h^*}$ behaves as follows. Given some root root (level h of some tree) the algorithm finds by the collision freeness of G the left and right pre-image nodes of the tree emerging from that root. This is repeated recursively until all pre-images of all leaves are returned or the maximal depth $h^* = \lceil \log(\eta\nu\mu) \rceil$ has been reached. Note that by the collision freeness of G, every node in any tree which is the output of algorithm $\mathsf{HashTree}$ has a unique pre-image. By the chain freeness, no adversary is able to span a tree which contains cycles. Therefore, the algorithm terminates. We assume w.l.o.g. that algorithm $\mathsf{PreimageLeafs}_{h^*}(\mathsf{root}, 0)$ always returns exactly $\eta' = \eta\nu\mu$ leaves, otherwise it is padded with arbitrary leaves.

Consider a signature $\boldsymbol{\sigma}_i = (\boldsymbol{c}'_i, \boldsymbol{s}'_i, \mathbf{auth}_i)$ on message $\boldsymbol{m}_i$ output by A. It remains to show that a valid signature leads to a valid transcript in the $\mathbf{OMMIM}$ experiment, i.e,. $\boldsymbol{b}_i = 1$ in Line 09. By the validity of the signature, $\boldsymbol{c}'_i = \mathsf{H}(\mathsf{RootCalc}(\mathsf{F}(\boldsymbol{s}'_i) - \boldsymbol{c}'_i \cdot pk, \mathbf{auth}_i), \boldsymbol{m}_i) = \mathsf{H}(\mathsf{RootCalc}(\boldsymbol{R}'_i, \mathbf{auth}_i), \boldsymbol{m}_i) = \mathsf{H}(\mathbf{root}_i, \boldsymbol{m}_i)$, where $\mathbf{root}_i := \mathsf{RootCalc}(\boldsymbol{R}'_i, \mathbf{auth}_i)$. By correctness of algorithm $\mathsf{PreimageLeafs}_{h^*}$, we have $\mathsf{H}(\mathbf{root}_i, \boldsymbol{m}_i) = \mathtt{Ch}(\mathsf{PreimageLeafs}_{h^*}(\mathbf{root}_i, 0)) = \mathtt{Ch}(\boldsymbol{R}'_{1,1,1}, \ldots, \boldsymbol{R}'_{\eta,\nu,\mu})$. Since all messages in $\boldsymbol{m}$ are distinct, each $\boldsymbol{c}'_i = \mathsf{H}(\mathbf{root}_i, \boldsymbol{m}_i)$ is distinct and thus every signature corresponds to a distinct session with oracle $\mathtt{Ch}$.

Therefore, B can make a successful query to oracle $\mathtt{V}_2(vSid, \boldsymbol{s}_i')$ in line 09 resulting in $\boldsymbol{b}_i = 1$ for every valid signature. Since overall, B makes $\ell(\mathsf{B}) = Q_{\mathsf{P}_2}(\mathsf{B}) + 1$ successful queries to $\mathtt{V}_2$, B wins $\mathbf{OMMIM}_{\mathsf{ID},par}$ whenever A wins $\mathbf{OMUF}'_{\mathsf{BS},par}$. This proves $\varepsilon' \geq \varepsilon - \frac{Q_{\mathsf{G}}^2}{2^{2\lambda}} - \frac{Q_{\mathsf{G}}}{2^{2\lambda}}$. Moreover, the number of abandoned sessions (denoted as $Q_{\mathsf{S}_1}(\mathsf{A})$) in the $\mathbf{OMUF}_{\mathsf{BS},par}$ experiment equals the number of abandoned sessions (denoted as $Q_{\mathsf{P}_1}(\mathsf{B})$) in the $\mathbf{OMMIM}_{\mathsf{ID},par}$ experiment and the number $Q_{\mathsf{V}}(\mathsf{B})$ of calls to oracle $\mathtt{Ch}$ is bounded by Q_{H} (for the simulation of H) plus additional $Q_{\mathsf{P}_2}(\mathsf{A}) + 1$ calls in Line 07 (the latter calls are necessary in case A guesses the output of $\mathtt{Ch}$ on some points). Finally, the running times of A and B are roughly the same, i.e. $t \approx t'$.

```
Adversary $\mathsf{B}^{\mathtt{P}_1,\mathtt{P}_2,\mathtt{V}_1,\mathtt{V}_2}(pk)$:
01 $h^* \leftarrow \lceil \log(\eta') \rceil$
02 $((\boldsymbol{m}_1, \boldsymbol{\sigma}_1), ..., (\boldsymbol{m}_\ell, \boldsymbol{\sigma}_\ell)) \leftarrow \mathsf{A}^{\mathsf{S}_1,\mathsf{S}_2,\mathsf{G},\mathsf{H}}(pk)$
03 For $i \in [\ell]$ do:
04     $(\boldsymbol{c}_i', \boldsymbol{s}_i', \mathbf{auth}_i) \leftarrow \boldsymbol{\sigma}_i$
05     $\boldsymbol{R}_i' \leftarrow \mathsf{F}(\boldsymbol{s}_i') - \boldsymbol{c}_i' \cdot pk$
06     $\mathbf{root}_i \leftarrow \mathsf{RootCalc}(\boldsymbol{R}_i', \mathbf{auth}_i)$
07     $\mathsf{H}(\mathbf{root}_i, \boldsymbol{m}_i)$
08     $vSid \leftarrow \mathbf{vSess}_{\mathbf{root}_i, \boldsymbol{m}_i}$
09     $\boldsymbol{b}_i \leftarrow \mathtt{V}_2(vSid, \boldsymbol{s}_i')$
10

Algorithm $\mathsf{S}_1$:
11 $(pSid, \boldsymbol{R}) \xleftarrow{\$} \mathtt{P}_1$
12 $(\boldsymbol{R}_1, \dots, \boldsymbol{R}_{\eta'}) \leftarrow \boldsymbol{R}$
13 Return $(pSid, \boldsymbol{R}_1, \dots, \boldsymbol{R}_\eta)$

Algorithm $\mathsf{S}_2(sid, c)$:
14 $pSid \leftarrow sid$
15 $\boldsymbol{s}_{\boldsymbol{pSid}} \leftarrow \mathtt{P}_2(pSid, c)$
16 Return $\boldsymbol{s}_{pSid}$

Algorithm $\mathsf{PreimageLeafs}_{h^*}(\mathsf{root}, h)$
17 If $(\exists (l, r) : \mathsf{G}(l, r) = \mathsf{root}: ) \wedge (h \leq h^*)$
18     Return $(\mathsf{PreimageLeafs}_{h^*}(l, h+1),$
               $\mathsf{PreimageLeafs}_{h^*}(r, h+1))$
19 Else
20     Return root

Algorithm $\mathsf{G}(e)$:
21 If $\boldsymbol{G}_e \neq \bot$: Return $\boldsymbol{G}_e$
22 $\boldsymbol{G}_e \xleftarrow{\$} \{0,1\}^{2\lambda}$
23 If $\exists e' \neq e$ s.t. $\boldsymbol{G}_e = \boldsymbol{G}_{e'}$:        //no collisions
24     Abort
25 If $\boldsymbol{G}_{\boldsymbol{G}_e} \neq \bot$:        //chain free
26     Abort
27 Return $\boldsymbol{G}_e$

Algorithm $\mathsf{H}(\mathsf{root}, m)$:
28 if $\boldsymbol{H}_{\mathsf{root},m} \neq \bot$: Return $\boldsymbol{H}_{\mathsf{root},m}$
29 $(\boldsymbol{R}_1', \dots, \boldsymbol{R}_{\eta'}') \leftarrow \mathsf{PreimageLeafs}_{h^*}(\mathsf{root}, 0)$
30 $(vSid, \boldsymbol{c}') \xleftarrow{\$} \mathtt{Ch}(\boldsymbol{R}_1', \dots, \boldsymbol{R}_{\eta'}')$
31 $\mathbf{vSess}_{\mathsf{root},m} \leftarrow vSid$
32 $\boldsymbol{H}_{\mathsf{root},m} \leftarrow \boldsymbol{c}'$
33 Return $\boldsymbol{H}_{\mathsf{root},m}$
```

Fig. 8. Construction of adversary B in the $\mathbf{OMMIM}_{\mathsf{ID},par}$ experiment from adversary A in the $\mathbf{OMUF}'_{\mathsf{BS},par}$ experiment.

We will now state that $\mathsf{ID}_{\eta'}[\mathsf{LHF}]$ is OMMIM secure.

Theorem 4. *Let* LHF *be a* $(\varepsilon, \eta' Q_{\mathsf{P}_1})$*-regular linear hash function family with a torsion-free element from the kernel. If* LHF *is* (ε', t')-**CR** *relative to par* $\in \mathsf{PGen}(1^\kappa)$ *then* $\mathsf{ID}_{\eta'}[\mathsf{LHF}]$ *is* $(\varepsilon, t, Q_{\mathsf{V}}, Q_{\mathsf{P}_1}, Q_{\mathsf{P}_2})$-**OMMIM** *relative to par, where*

$$t' = 2t, \quad \varepsilon' = O\left(\left(\varepsilon^2 - \frac{(Q_{\mathsf{V}} Q_{\mathsf{P}_1})^{Q_{\mathsf{P}_2}+1}}{\left(\frac{|\mathcal{C}|}{|\mathcal{S}_c|}\right)^{Q_{\mathsf{P}_2}} \cdot |\mathcal{C}|}\right)^2 \frac{1}{Q_{\mathsf{V}}^2 Q_{\mathsf{P}_2}^3}\right).$$

Proof. The technical proof of this theorem will be given in Appendix A.

6 Instantiation from Lattices

We now give a lattice-based example of a LHF with noticeable correctness error which is derived from Lyubashevsky's identification scheme [38] and has also been implicitly used in [56].

NOTATION. Let R and R_q denote the rings $\mathbb{Z}[X]/\langle X^n + 1 \rangle$ and $\mathbb{Z}_q[X]/\langle X^n + 1 \rangle$, for integer $n = 2^r$, where $r \in \mathbb{Z}^+$ and q is an odd integer. Polynomials in R_q have degree at most $n-1$ and coefficients in range $[-(q-1)/2, (q-1)/2]$.

For such coefficients we abuse the notation *mod* to denote with $x' = x \bmod q$, the unique element x' s.t. for any integer k: $x' = x + kq$ and $x' \in [-(q-1)/2, (q-1)/2]$. Bold lower-case letters denote elements in R_q and bold lower-case letters with a hat denote vectors of vectors with coefficients in ring R_q. To measure the size of elements $\mathbf{x} = x_0 + x_1X^1 + \cdots + x_{n-1}X^{n-1}$ in ring R_q we define norm p_∞ as $\|\mathbf{x}\|_\infty := \max_i |x_i \mod q|$. In rings R and R_q, $\|x_i\|_\infty$ represents $|x_i|$ and $|x_i \bmod q|$, respectively. Similarly, for $\hat{\mathbf{x}} = (\mathbf{x}_0, \ldots, \mathbf{x}_{k-1})$, we define norm p_∞ as $\|\hat{\mathbf{x}}\|_\infty := \max_i \|\mathbf{x}_i\|_\infty$. Further we define the p_1 norm as $\|\mathbf{x}\|_1 := \sum_i |\mathbf{x}_i|$ and p_2 norm as $\|\mathbf{x}\|_2 := (\sum_i |\mathbf{x}_i|^2)^{1/2}$. It is not hard to see that for any two polynomials $\mathbf{e}, \mathbf{f} \in R_q$,

$$\|\mathbf{e} \cdot \mathbf{f}\|_\infty \leq \|\mathbf{e}\|_\infty \|\mathbf{f}\|_1 \leq n \|\mathbf{e}\|_\infty \|\mathbf{f}\|_\infty \ . \tag{4}$$

We now recall the $\mathsf{R\text{-}SIS}_{q,n,m,d}$ problem over R_q [48,40].

Definition 6 **($\mathsf{R\text{-}SIS}_{q,n,m,d}$).** *We say that* $\mathsf{R\text{-}SIS}_{q,n,m,d}$ *is (ε, t)-hard if for all adversaries* A *running in time at most t, the probability that $\mathsf{A}(\hat{\boldsymbol{a}})$ (where $\hat{\boldsymbol{a}} \xleftarrow{\$} R_q^m$) outputs a non-zero $\hat{\boldsymbol{z}} \in R_q^m$ s.t. $\sum_{i=1}^m \mathbf{a}_i \cdot \mathbf{z}_i = 0$ and $\|\hat{\boldsymbol{z}}\|_\infty \leq d$, is bounded by ε.*

Similarly, $\mathsf{R\text{-}SIS}_{q,n,m,d}$ *is (ε, t)-hard relative to $\hat{\boldsymbol{a}} \in R_q^m$ if for all adversaries* A *running in time at most t, the probability that* A *outputs a non-zero $\hat{\boldsymbol{z}} \in R_q^m$ s.t. $\sum_{i=1}^m \mathbf{a}_i \cdot \mathbf{z}_i = 0$ and $\|\hat{\boldsymbol{z}}\|_\infty \leq d$, is bounded by ε.*

Let us first estimate the concrete hardness of solving the $\mathsf{R\text{-}SIS}_{q,n,m,d}$ problem for uniformly random $\hat{\boldsymbol{a}}$ which is equivalent to finding a short vector in the related lattice

$$\mathbf{\Lambda}_q^\perp(\hat{\mathbf{a}}) = \{\hat{\mathbf{z}} \in R_q^m : \sum_{i=1}^m \mathbf{a}_i \cdot \mathbf{z}_i = 0\}.$$

Gama and Nguyen [28] classified algorithms for finding short vectors in random lattices in terms of the root Hermite factor δ. Such algorithms compute a vector of length δ^n times the shortest vector of the lattice. Whereas $\delta = 1.01$ can be achieved, it is conjectured that a factor of $\delta = 1.007$ may not be achievable [22].

We use the following estimation from [38, Eqn. 3] to estimate the length of the shortest vector (in p_∞ norm) which can be efficiently found in lattice $\mathbf{\Lambda}_q^\perp(\hat{\boldsymbol{a}})$ as

$$\mathsf{sv}_\delta(n, q) := \min\{q, 2^{2\sqrt{n \log(q) \log(\delta)}}(n \log(q)/\log(\delta))^{-1/4}\}.$$

We make the following conjecture about $\mathsf{R\text{-}SIS}_{q,n,m,d}$ with $\delta = 1.005$.

Conjecture 1. If $d < \mathsf{sv}_{1.005}(n, q)$ then no efficient algorithm can solve $\mathsf{R\text{-}SIS}_{q,n,m,d}$.

We note that security of our blind signature scheme depends on the hardness of $\mathsf{R\text{-}SIS}_{q,n,m,d}$ relative to fixed $\hat{\boldsymbol{a}}$. However, as discussed in Section 2, our theorems can be easily re-written to work in a setting where $\hat{\boldsymbol{a}} \xleftarrow{\$} R_q^m$ is chosen uniformly at random.

LINEAR HASH FUNCTION. We select the parameters according to Figure 9. Firstly, variables q, n specify the ring $R_q := \mathbb{Z}_q[X]/\langle X^n + 1\rangle$, where n is a power of two. Define the sets

$$\mathcal{S} := R_q, \mathcal{D} := R_q^m, \text{ and } \mathcal{R} := R_q,$$

For the hardness of collision resistance, we select d such that $d < \frac{1}{2}\mathsf{sv}_{1.005}(n, q)$ and also $\mathcal{D}' \subseteq \mathcal{B}_q(d)$ [6] where $\mathcal{B}_q(w)$ is defined as

$$\mathcal{B}_q(w) := \{\mathbf{s} \in R_q : \|\mathbf{s}\|_\infty \leq w\}.$$

For $\hat{\boldsymbol{e}}, \hat{\boldsymbol{f}} \in \mathcal{D}$ and $\mathbf{g} \in \mathcal{S}$ we define addition $\hat{\boldsymbol{e}} + \hat{\boldsymbol{f}} := (\mathbf{e}_1 + \mathbf{f}_1, \ldots, \mathbf{e}_m + \mathbf{f}_m)$, multiplication $\hat{\mathbf{e}} \cdot \hat{\mathbf{f}} := (\mathbf{e}_1\mathbf{f}_1, \ldots, \mathbf{e}_m\mathbf{f}_m)$, and scalar multiplication $\mathbf{g} \cdot \hat{\boldsymbol{e}} = (\mathbf{g}\mathbf{e}_1, \ldots, \mathbf{g}\mathbf{e}_m)$. This makes $\mathcal{R}$ and $\mathcal{D}$ modules over $\mathcal{S}$.

Algorithm $\mathsf{PGen}(1^\kappa)$ returns a random element $par = \hat{\boldsymbol{a}} \xleftarrow{\$} \mathcal{R}^m$. Algorithm $\mathsf{F} : \mathcal{D} \mapsto \mathcal{R}$ is defined for any $\hat{\boldsymbol{z}} \in \mathcal{D}$ as,

$$\mathsf{F}(\hat{\boldsymbol{z}}) := \sum_{i=1}^m \mathbf{a}_i \cdot \mathbf{z}_i \bmod q \ .$$

Clearly, F is a module homomorphism since for every $\hat{\boldsymbol{y}}, \hat{\boldsymbol{z}} \in \mathcal{D}$, $\mathbf{c} \in \mathcal{R}$: $\mathsf{F}(\hat{\boldsymbol{y}} + \hat{\boldsymbol{z}}) = \hat{\boldsymbol{a}}(\hat{\boldsymbol{y}} + \hat{\boldsymbol{z}}) = \hat{\boldsymbol{a}}\hat{\boldsymbol{y}} + \hat{\boldsymbol{a}}\hat{\boldsymbol{z}} = \mathsf{F}(\hat{\boldsymbol{y}}) + \mathsf{F}(\hat{\boldsymbol{z}})$ and $\mathsf{F}(\hat{\boldsymbol{y}}\mathbf{c}) = \hat{\boldsymbol{a}}(\mathbf{y}_1\mathbf{c}, \ldots, \mathbf{y}_m\mathbf{c}) = \mathbf{a}_1\mathbf{y}_1\mathbf{c} + \cdots + \mathbf{a}_m\mathbf{y}_m\mathbf{c} = \mathsf{F}(\hat{\boldsymbol{y}})\mathbf{c}$.

Parameter	Definition	Instantiation
n	integer that is power of 2	1024
m	dimension of a secret key vector	200
q	prime	2^{3550}
ι	# of irreducible factors of $X^n + 1$ modulo q	64
δ	p_∞ of a torsion-free element from the kernel	2^{19}
d_{sk}	p_∞ of a secret key	2^{169}
$d_{c'}$	p_∞ of an output of a random oracle	2^{85}
u	integer	4
v	integer	4
w	integer	4
μ	number of $\boldsymbol{\beta}_j$	60
η	number of $\mathcal{R}_i$	60
ν	number of $\hat{\boldsymbol{\alpha}}_k$	60
d_β	$ud_{c'}n$	2^{97}
d_c	$d_\beta - d_{c'}$	2^{97}
d_r	$\geq vmn^2 d_{sk} d_c$	2^{295}
d_s	$d_r - nd_{sk}d_c$	2^{295}
d_α	$wd_s nm$	2^{315}
$d_{s'}$	$d_\alpha - d_s$	2^{315}
d	$d < \frac{1}{2}\mathsf{sv}_{1.005}(n, q)$	2^{316}
sig size		7.73 MB

Fig. 9. Definition of parameters for the lattice-based LHF.

For $\mathsf{xxx} \in \{\beta, c, c'\}$ and $\mathsf{yyy} \in \{sk, r, s, s', \alpha\}$, the filter sets are defined as

$$\mathcal{S}_{\mathsf{xxx}} := \mathcal{B}_q(d_{\mathsf{xxx}}) \subseteq \mathcal{S}, \quad \mathcal{D}_{\mathsf{yyy}} := \mathcal{B}_q^m(d_{\mathsf{yyy}}) \subseteq \mathcal{D}.$$

To estimate the membership of sums and products of $\hat{\boldsymbol{e}}, \hat{\boldsymbol{f}} \in \mathcal{D}$ to specific subsets of $\mathcal{D}$ we use the lemma proven by Rückert [56].

Lemma 2. *Let k, d_a, d_b and γ be integers, s.t. $d_b \geq \gamma k n d_a$. Then, for all $\hat{\mathbf{a}} \in \mathcal{B}_q^k(d_a)$,*

$$\Pr_{\hat{\mathbf{b}} \xleftarrow{\$} \mathcal{B}_q^k(d_b)} \left[\left\| \hat{\mathbf{a}} + \hat{\mathbf{b}} \right\|_\infty \leq d_b - d_a \right] > e^{-1/\gamma} - o(1) \ .$$

Enclosedness Errors and Smoothness. First, we focus on calculating the enclosedness errors of LHF based on parameters chosen in Figure 9. Later on, we also show that LHF is smooth.

Lemma 3. *If $d_\beta, d_r, d_\alpha, d_c, d_s, d_{s'}$ are defined as in Figure 9 and* LHF *is defined as above, then* LHF *has enclosedness errors equal to:*

$$\left(1 - e^{-1/u} + o(1), 1 - e^{-1/v} + o(1), 1 - e^{-1/w} + o(1)\right) .$$

Proof. The statement follows straightforwardly from Lemma 2 and the way we picked $d_\beta, d_r, d_\alpha, d_c, d_s, d_{s'}$. □

In Figure 9 we select $u = v = w$. Thus, we need choose appropriate μ, η, ν to make sure that correctness error of our blind signature is negligible. Indeed, we simply pick $\mu = \eta = \nu$ such that

$$(1 - e^{-1/u} + o(1))^\mu < 2^{-130}.$$

Then by Lemma 1, BS[LHF] has correctness error at most $3 \cdot 2^{-130} < 2^{-128}$.

Lemma 4. *If d_s and d_c are defined as in Figure 9, then* LHF *is smooth.*

[6] We recall that the set $\mathcal{D}'$ is defined in Equation 2.

Proof. In the following we prove smoothness conditions (S1) and (S2). Condition (S3) can be proven analogously to (S2). Conditions (S4), (S5) and (S6) can be proven analogously to (S1), (S2) and (S3), respectively.

Since $d_\alpha = d_s + d_{s'}$, for all $\hat{\boldsymbol{s}} \in \mathcal{D}_s$ and $\hat{\boldsymbol{s}}' \in \mathcal{D}_{s'}$, $\|\hat{\boldsymbol{s}}' - \hat{\boldsymbol{s}}\|_\infty \leq d_{s'} + d_s = d_\alpha$ and therefore $\hat{\boldsymbol{s}}' - \hat{\boldsymbol{s}} \in \mathcal{D}_\alpha$. This proves smoothness condition (S1).

To prove (S2), we fix $\hat{\boldsymbol{s}}_1, \hat{\boldsymbol{s}}_2 \in \mathcal{D}_s$ and define sets $\mathcal{D}_{\alpha_1} := \{\hat{\boldsymbol{\alpha}} \in \mathcal{D}_\alpha \mid \hat{\boldsymbol{\alpha}} + \hat{\boldsymbol{s}}_1 \in \mathcal{D}_{s'}\}$ and $\mathcal{D}_{\alpha_2} := \{\hat{\boldsymbol{\alpha}} \in \mathcal{D}_\alpha \mid \hat{\boldsymbol{\alpha}} + \hat{\boldsymbol{s}}_2 \in \mathcal{D}_{s'}\}$. Note that for all $\hat{\boldsymbol{s}}_1, \hat{\boldsymbol{s}}_2 \in \mathcal{D}_s$ and $\hat{\boldsymbol{s}}' \in \mathcal{D}_{s'}$ there exist $\hat{\boldsymbol{\alpha}}_1 \in \mathcal{D}_{\alpha_1}$ and $\hat{\boldsymbol{\alpha}}_2 \in \mathcal{D}_{\alpha_2}$ s.t. $\hat{\boldsymbol{\alpha}}_1 + \hat{\boldsymbol{s}}_1 = \hat{\boldsymbol{s}}'$ and $\hat{\boldsymbol{\alpha}}_2 + \hat{\boldsymbol{s}}_2 = \hat{\boldsymbol{s}}'$. So, $|\mathcal{D}_{\alpha_1}| = |\mathcal{D}_{\alpha_2}| = |\mathcal{D}_{s'}|$.

In the following, fix $\hat{\boldsymbol{s}}_1, \hat{\boldsymbol{s}}_2 \in \mathcal{D}_s$ and define the random variables $\hat{\boldsymbol{\alpha}}' \xleftarrow{\$} \mathcal{D}_{\alpha_2}$ and $\hat{\boldsymbol{\alpha}}^* \xleftarrow{\$} \mathcal{D}_{\alpha_1}$. To prove smoothness condition (S2), it remains to show that

$$\Delta(\hat{\boldsymbol{\alpha}}', \hat{\boldsymbol{\alpha}}^* + \hat{\boldsymbol{s}}_1 - \hat{\boldsymbol{s}}_2) = 0, \tag{5}$$

We have

$$\begin{aligned}\Delta(\hat{\boldsymbol{\alpha}}', \underline{\hat{\boldsymbol{\alpha}}}) =& \frac{1}{2} \sum_{\bar{\hat{\boldsymbol{\alpha}}} \notin \mathcal{D}_{\alpha_2}} \left| \Pr_{\hat{\boldsymbol{\alpha}}' \xleftarrow{\$} \mathcal{D}_{\alpha_2}} [\hat{\boldsymbol{\alpha}}' = \bar{\hat{\boldsymbol{\alpha}}}] - \Pr_{\hat{\boldsymbol{\alpha}}^* \xleftarrow{\$} \mathcal{D}_{\alpha_1}} [\hat{\boldsymbol{\alpha}}^* + \hat{\boldsymbol{s}}_1 = \bar{\hat{\boldsymbol{\alpha}}} + \hat{\boldsymbol{s}}_2] \right| \\ =& \frac{1}{2} \sum_{\bar{\hat{\boldsymbol{\alpha}}} \in \mathcal{D}_{\alpha_2}} \left| \Pr_{\hat{\boldsymbol{\alpha}}' \xleftarrow{\$} \mathcal{D}_{\alpha_2}} [\hat{\boldsymbol{\alpha}}' = \bar{\hat{\boldsymbol{\alpha}}}] - \Pr_{\hat{\boldsymbol{\alpha}}^* \xleftarrow{\$} \mathcal{D}_{\alpha_1}} [\hat{\boldsymbol{\alpha}}^* + \hat{\boldsymbol{s}}_1 = \bar{\hat{\boldsymbol{\alpha}}} + \hat{\boldsymbol{s}}_2] \right| .\end{aligned} \tag{6}$$

To show that (6) amounts to zero, we argue as follows. If $\bar{\hat{\boldsymbol{\alpha}}} \notin \mathcal{D}_{s'}$ then clearly

$$\Pr_{\hat{\boldsymbol{\alpha}}' \xleftarrow{\$} \mathcal{D}_{\alpha_2}} [\hat{\boldsymbol{\alpha}}' = \bar{\hat{\boldsymbol{\alpha}}}] = 0 = \Pr_{\hat{\boldsymbol{\alpha}}^* \xleftarrow{\$} \mathcal{D}_{\alpha_1}} [\hat{\boldsymbol{\alpha}}^* + \hat{\boldsymbol{s}}_1 = \bar{\hat{\boldsymbol{\alpha}}} + \hat{\boldsymbol{s}}_2].$$

Now, suppose $\bar{\hat{\boldsymbol{\alpha}}} \in \mathcal{D}_{s'}$. Since $\hat{\boldsymbol{\alpha}}' \in \mathcal{D}_{\alpha_2}$ and random variable $\hat{\boldsymbol{\alpha}}'$ takes values in $\mathcal{D}_{\alpha_2}$, the probability that random variable $\hat{\boldsymbol{\alpha}}'$ takes value $\bar{\hat{\boldsymbol{\alpha}}}$ is $\frac{1}{|\mathcal{D}_{\alpha_2}|} = \frac{1}{|\mathcal{D}_{s'}|}$. So, $\Pr_{\hat{\boldsymbol{\alpha}}' \in \mathcal{D}_{\alpha_2}} [\hat{\boldsymbol{\alpha}}' = \bar{\hat{\boldsymbol{\alpha}}}] = \frac{1}{|\mathcal{D}_{s'}|}$. Since $\bar{\hat{\boldsymbol{\alpha}}} \in \mathcal{D}_{\alpha_2}$, $\bar{\hat{\boldsymbol{\alpha}}} + \hat{\boldsymbol{s}}_2 \in \mathcal{D}_{s'}$. Also $\hat{\boldsymbol{\alpha}}^* \in \mathcal{D}_{\alpha_1}$ implies $\hat{\boldsymbol{\alpha}}^* + \hat{\boldsymbol{s}}_1 \in \mathcal{D}_{s'}$. So the probability that random variable $\hat{\boldsymbol{\alpha}}^*$ fulfills $\hat{\boldsymbol{\alpha}}^* + \hat{\boldsymbol{s}}_1 = \bar{\hat{\boldsymbol{\alpha}}} + \hat{\boldsymbol{s}}_2$ is $\frac{1}{|\mathcal{D}_{\alpha_1}|} = \frac{1}{|\mathcal{D}_{s'}|}$. Therefore, $\Pr_{\hat{\boldsymbol{\alpha}}^* \in \mathcal{D}_{\alpha_1}} [\hat{\boldsymbol{\alpha}}^* + \hat{\boldsymbol{s}}_1 = \bar{\hat{\boldsymbol{\alpha}}} + \hat{\boldsymbol{s}}_2] = \frac{1}{|\mathcal{D}_{s'}|}$. This completes the proof.

TORSION FREE ELEMENTS FROM THE KERNEL. We first observe that we only need to find a non-zero $\hat{\mathbf{z}}^*$ such that $\mathsf{F}(\hat{\mathbf{z}}^*) = \mathbf{0}$. Indeed, if d_c is small enough then by selecting appropriate prime q we can apply the main result of Lyubashevsky and Seiler [41].

Lemma 5 ([41] Corollary 1.2). *Let $n \geq \iota > 1$ be powers of 2 and $q \equiv 2\iota + 1 \pmod{4\iota}$ be a prime. Then $X^n + 1$ factors into ι irreducible polynomials $X^{n/\iota} - r_j$ modulo q and any $\mathbf{y} \in R_q \setminus \{\mathbf{0}\}$ that satisfies*

$$\|\mathbf{y}\|_\infty < \frac{1}{\sqrt{\iota}} \cdot q^{1/\iota} \quad \textit{or} \quad \|\mathbf{y}\|_2 < q^{1/\iota}$$

is invertible in R_q.

Hence, pick $d_c < \frac{1}{2\sqrt{\iota}} \cdot q^{1/\iota}$. Then, for $\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{S}_c$, $(\mathbf{c}_1 - \mathbf{c}_2)\hat{\mathbf{z}}^* = \mathbf{0} \implies \mathbf{c}_1 = \mathbf{c}_2$ since otherwise $\mathbf{c}_1 - \mathbf{c}_2$ is invertible and thus $\hat{\mathbf{z}}^* = \mathbf{0}$. Therefore, $\hat{\mathbf{z}}^*$ is a torsion-free element from the kernel.

Many papers investigate non-existence of a short vector in random module lattices e.g. [36,43]. However, here we are interested in the existence. Concretely, we want to make sure there exists a $\hat{\mathbf{z}}^*$ from the kernel with infinity norm at most $\delta < q/2$. Consider the set of vectors $B_\delta \subset R_q^m$ of polynomials with coefficients between 0 and δ. Clearly, for $\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2 \in B_\delta$: $\|\hat{\mathbf{y}}_1 - \hat{\mathbf{y}}_2\|_\infty \leq \delta < q/2$. If we select δ such that $|B_\delta| = (\delta + 1)^{nm} > q^n$ then by the pigeonhole principle, there exist two distinct $\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2 \in B_\delta$ such that $\mathsf{F}(\hat{\mathbf{y}}_1) = \mathsf{F}(\hat{\mathbf{y}}_2)$. Hence, we can set $\hat{\mathbf{z}}^* = \hat{\mathbf{y}}_1 - \hat{\mathbf{y}}_2$.

COLLISION RESISTANCE. To estimate the hardness of finding collisions in LHF we state the following simple lemma.

Lemma 6. *If R-SIS$_{q,n,m,2d}$ is (ε, t)-hard relative to $\hat{\mathbf{a}} \in R_q^m$ then* LHF *is (ε, t)-***CR** *relative to $par \in$* PGen, *where par contains all the values defined in Fig. 9 along with $\hat{\mathbf{a}}$.*

Proof. Adversary A returns distinct values $\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2 \in \mathcal{D}'$ after being called on parameters par. Since $\mathsf{F}(\hat{\boldsymbol{x}}_1) = \mathsf{F}(\hat{\boldsymbol{x}}_2)$ and since F is a module homomorphism, $\mathsf{F}(\hat{\boldsymbol{x}}_2 - \hat{\boldsymbol{x}}_1) = \mathsf{F}(\hat{\boldsymbol{x}}_2) - \mathsf{F}(\hat{\boldsymbol{x}}_1) = 0$. Further, $\|\hat{\boldsymbol{x}}_2 - \hat{\boldsymbol{x}}_1\|_\infty \leq 2d$. So $\hat{\boldsymbol{x}}_2 - \hat{\boldsymbol{x}}_1$ is a solution to the $\mathsf{R}\text{-}\mathsf{SIS}_{q,n,m,2d}$ problem relative to $\hat{\boldsymbol{a}}$.

As we described in Section A, adversary A manages to extract $\hat{\chi}_1, \hat{\chi}_2$ so that $\mathsf{F}(\hat{\chi}_1 - \hat{\chi}_2) = 0$. The norm of $\hat{\chi}_1$ (and similarly for $\hat{\chi}_2$) can be simply bounded by:

$$\|\hat{\chi}_1\|_\infty \leq d_{s'} + nd_{c'}d_{sk} < 2d_{s'}.$$

Thus, we set $d = 2d_{s'}$. With parameters defined in Figure 9, $d \approx 2^{233}$ and $\frac{1}{2}\mathsf{sv}_{1.005}(n,q) \approx 2^{235}$. Therefore, we get $\|\hat{\chi}_1 - \hat{\chi}_2\|_\infty < 2d < \mathsf{sv}_{1.005}(n,q)$.

ONE-MORE UNFORGEABILITY. In order for the probability in Lemma 13 to be negligible, we need to set large enough output space $\mathcal{S}_{c'}$ of the random oracle. For concreteness, we define $Q_{\mathsf{V}} = Q_{\mathsf{P}_1} = 2^{128}$ and $Q_{\mathsf{P}_2} = 7$ since we only allow seven signing queries due to a potential lattice variant of the ROS attack (see Section 7). For the parameters defined in Figure 9, the probability in Lemma 13 is around 2^{-128}.

REGULARITY. We now prove that by selecting sizes d_{sk} and d_r as in Figure 9, our LHF is (ϵ, Q')-regular where $\varepsilon = 2^{-128}$ and $Q' = Q_{\mathsf{P}_2}\mu\eta\nu$.

Lemma 7. *Denote $\varepsilon = 2^{-128}$ and $Q' = 7\mu\eta\nu$. Then, for our selection of d_{sk}, d_r, the LHF is (ϵ, Q')-regular, i.e.*

$$\frac{|\mathcal{D}'_{sk}|}{|\mathcal{D}_{sk}|} \cdot \left(\frac{|\mathcal{D}'_r|}{|\mathcal{D}_r|}\right)^{Q'} \geq 1 - 2^{-130} = 1 - \varepsilon/4, \tag{7}$$

where

$$\mathcal{D}'_{sk} := \{\hat{\boldsymbol{sk}} \in \mathcal{D}_{sk} : \hat{\boldsymbol{sk}} + \hat{\mathbf{z}}^* \in \mathcal{D}_{sk}\}$$

and

$$\mathcal{D}'_r := \{\hat{\boldsymbol{r}} \in \mathcal{D}_r : \forall \mathbf{c} \in \mathcal{S}_c, \hat{\boldsymbol{r}} + \mathbf{c}\hat{\mathbf{z}}^* \in \mathcal{D}_r\}.$$

Proof. Indeed, we first picked d_r so that

$$\left(\frac{|\mathcal{D}'_r|}{|\mathcal{D}_r|}\right)^{Q'} \geq 1 - 2^{-131}.$$

Simultaneously, we chose d_{sk} which satisfies: $|\mathcal{D}'_{sk}|/|\mathcal{D}_{sk}| \geq 1 - 2^{-131}$. Also, we check that $d_r \geq vmn^2 d_{sk} d_c$ for the enclosedness property. Then, Equation (7) follows by the Bernoulli inequality.

SIZES. We pick prime $q \approx 2^{3550}$ so that $q \equiv 2\iota + 1 \pmod{4\iota}$ where $\iota = 64$ and $X^n + 1$ splits into ι irreducible polynomials modulo q. Hence, we can apply Lemma 5. Unfortunately, such a large prime modulus affects the signing time significantly. The signature consists of three parts: $\hat{\boldsymbol{s}}', \mathbf{c}'$ and auth. The size for $\hat{\boldsymbol{s}}'$ and $\mathbf{c}'$ are respectively $nm \log 2d_{s'}$ and $n \log 2d_{c'}$. Also, auth contains the index of the leaf (which can be represented with at most $\log(\mu\eta\nu)$ bits) and $\log(\mu\eta\nu)$ outputs of the hash function G. If we assume that $\mathsf{G} : \{0,1\}^* \to \{0,1\}^{128}$ then auth has at most $\log(\mu\eta\nu) + 128 \cdot \log(\mu\eta\nu)$ bits. For parameters selected in Figure 9, our signature has size around 7.73 MB. We observe that the main reason of obtaining such large signatures is the size for d_r and d_{sk}, which should satisfy the regularity property.

7 Generalized ROS Problem

The standard ROS (Random inhomogenities in an Overdetermined, Solvable system of linear equations) problem was first introduced by Schnorr [58] in the context of blind signatures. If one can solve the ROS problem then one is also able to break the security of the Schnorr as well as the Okamoto-Schnorr and Okamoto-Gouillou-Quisquarter blind signature schemes. Later works by Wagner and Minder and Sinclair [60,42] proposed algorithms which solve the ROS problem in sub-exponential time. In this section, we discuss main challenges when translating the ROS problem to general linear hash function families with correctness error. To the best of our knowledge, none of previous works on lattice-based blind signatures (e.g. [56,4,5]) consider this issue.

```
Game $\ell\text{-}\mathbf{GROS}_{\mathsf{LHF},par}$:
$\left(\boldsymbol{c} \in \mathcal{S}_c^{\ell+1}, \mathbf{A} \in \left(\mathcal{X}_\ell^{\ell+1} \times \mathcal{S}_{c'}^{\ell+1}\right)\right) \xleftarrow{\$} \mathsf{A}^{\mathsf{H}(\cdot)}(par)$
If $(\boldsymbol{c}_{\ell+1} = -1) \wedge (\mathbf{A}\boldsymbol{c} = 0) \wedge (\forall i, j \in [\ell+1] : \mathsf{H}(\mathbf{A}_{i,1}, \dots, \mathbf{A}_{i,\ell}) = \mathbf{A}_{i,\ell+1}) \wedge (\mathbf{A}_i \neq \mathbf{A}_j)$: Return 1
Return 0
```

Fig. 10. Game $\ell\text{-}\mathbf{GROS}_{\mathsf{LHF},par}$ with adversary A. $\mathsf{H}\colon \{0,1\}^* \to \mathcal{S}_{c'}$ is a random oracle.

We start by describing the Generalized ROS ($\mathbf{GROS}$) problem for linear hash function families with correctness error. For a linear hash function family LHF, $par \in \mathsf{PGen}$ and a positive integer ℓ, let $\mathcal{X}_\ell$ be the set

$$\mathcal{X}_\ell := \{(x_1, ..., x_\ell) \in \mathcal{S}^\ell \mid \forall \boldsymbol{s} \in \mathcal{D}_s^\ell : \boldsymbol{x} \cdot \boldsymbol{s} \in \mathcal{D}_{s'}\}. \tag{8}$$

The game $\ell\text{-}\mathbf{GROS}_{\mathsf{LHF},par}$ is defined via Figure 10. The advantage of adversary A in $\ell\text{-}\mathbf{GROS}_{\mathsf{LHF},par}$ is defined as $\mathbf{Adv}^{\ell\text{-}\mathbf{GROS}}_{\mathsf{LHF},par}(\mathsf{A}) := \Pr[\ell\text{-}\mathbf{GROS}^{\mathsf{A}}_{\mathsf{LHF},par} \Rightarrow 1]$ and its running time is denoted as $\mathbf{Time}^{\ell\text{-}\mathbf{GROS}}_{\mathsf{LHF},par}(\mathsf{A})$.

Definition 7 **(ℓ-$\mathbf{GROS}$ Hardness).** *Let $\ell \in \mathbb{N}, \ell > 0$ and let* LHF *be a linear function family and let* $par \in \mathsf{PGen}(1^\kappa)$*. ℓ-$\mathbf{GROS}$ is said to be $(\varepsilon, t, Q_{\mathsf{H}})$*-hard in the random oracle model relative to par and LHF *if for all adversaries A satisfying $\mathbf{Time}^{\ell\text{-}\mathbf{GROS}}_{\mathsf{LHF},par}(\mathsf{A}) \leq t$ and making at most Q_{H} queries to H, we have that $\mathbf{Adv}^{\ell\text{-}\mathbf{GROS}}_{\mathsf{LHF},par}(\mathsf{A}) \leq \varepsilon$.*

The following theorem shows that an attack on $\ell\text{-}\mathbf{GROS}_{\mathsf{LHF},par}$ propagates to an attack against $\mathbf{OMUF}_{\mathsf{BS}[\mathsf{LHF},\mathsf{G},\mathsf{H}],par}$.

Theorem 5. *Let LHF be a linear hash function family with enclosedness error $(\delta_1, \delta_2, \delta_3)$, $\mathsf{G}\colon \{0,1\}^* \to \{0,1\}^{2\lambda}$ and $\mathsf{H}\colon \{0,1\}^* \to \mathcal{S}_{c'}$ be random oracles and let $\mathsf{BS} := \mathsf{BS}_{\eta,\nu,\mu}[\mathsf{LHF}, \mathsf{G}, \mathsf{H}]$. If BS is $(\varepsilon, t, 0, \ell, Q_{\mathsf{G}}, Q_{\mathsf{H}})$-$\mathbf{OMUF}$ relative to par in the random oracle model then ℓ-$\mathbf{GROS}$ is $(\varepsilon/(1-\delta_2)^\ell, t, Q_{\mathsf{H}})$-hard relative to LHF and par in the random oracle model.*

Proof (Sketch). The proof is similar to one for perfect correctness [58,33]. For readability, we assume that $\mu = \nu = 1$ since there is no need to blind challenges/signatures in this scenario (it can, however, be easily generalized to arbitrary μ, ν). We now define an adversary B in $\mathbf{OMUF}_{\mathsf{BS},par}$ that internally runs an adversary A against $\ell\text{-}\mathbf{GROS}_{\mathsf{LHF},par}$ with random oracle H'.

- It simultaneously opens ℓ sessions with S_1, receiving commitments $\boldsymbol{R}_1, ..., \boldsymbol{R}_\ell$. Let us denote $\boldsymbol{R}_i = (\boldsymbol{R}_{i,1}, \dots, \boldsymbol{R}_{i,\eta})$ for $i \in [\ell]$.
- Next, it executes $\mathsf{A}^{\mathsf{H}'}(par)$. When A makes a fresh query $\boldsymbol{a}$ to H', B computes $\boldsymbol{R}'_{\boldsymbol{a},j} := \sum_{i=1}^{\ell} \boldsymbol{a}_i \cdot \boldsymbol{R}_{i,j}$ for all $j \in [\eta]$. It then computes $(\mathsf{root}_{\boldsymbol{a}}, \mathsf{tree}_{\boldsymbol{a}}) \leftarrow \mathsf{HashTree}(\boldsymbol{R}'_{\boldsymbol{a},1}, ..., \boldsymbol{R}'_{\boldsymbol{a},\eta})$ and $c' \leftarrow \mathsf{H}(\mathsf{root}_{\boldsymbol{a}}, m_{\boldsymbol{a}})$, for a fresh message $m_{\boldsymbol{a}}$, and then returns $\mathsf{H}'(\boldsymbol{a}) := c'$ as the answer. Clearly, c' is independent from commitments $\boldsymbol{R}_i$.
- When A terminates and returns $(\mathbf{A}, \boldsymbol{c})$, B sends the value $\boldsymbol{c}_i$ as the challenge value for the i^{th} session with S_2, where $i \in [\ell]$, and receives an answer $\boldsymbol{s}_i$. If $\boldsymbol{R}_{i,1} \neq \mathsf{F}(\boldsymbol{s}_i) - \boldsymbol{c}_i \cdot pk$ then B aborts. Note that the probability that B does not abort at all is at least $(1-\delta_2)^\ell$ by definition of the enclosedness error.
- Next, for all $j \in [\ell+1]$, B computes $\boldsymbol{s}'_j := \sum_{i=1}^{\ell} \boldsymbol{A}_{j,i} \cdot \boldsymbol{s}_i$ and retrieves the values $\mathsf{root}_{\boldsymbol{A}_j}, \mathsf{tree}_{\boldsymbol{A}_j}, m_{\boldsymbol{A}_j}$ used to compute $\boldsymbol{A}_j$ and computes $c'_j \leftarrow \mathsf{H}(\mathsf{root}_{\boldsymbol{A}_j}, m_{\boldsymbol{A}_j})$, $\mathsf{auth}_j \leftarrow \mathsf{BuildAuth}((0,1,1), \mathsf{tree})$. It sets $\sigma_j := (c'_j, \boldsymbol{s}'_j, \mathsf{auth}_j)$.
- Finally, B returns $\ell+1$ message/signature pairs $(\sigma_1, m_{\boldsymbol{A}_1}), ..., (\sigma_{\ell+1}, m_{\boldsymbol{A}_{\ell+1}})$.

Correctness of the signatures follows because

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

Further, by (8) we have $\boldsymbol{s}'_i \in \mathcal{D}_{s'}$ for all $i \in [\ell]$. Correctness of the authentication path can easily be verified. □

One observes that the attack only makes sense for small values of ℓ due to the security loss of $(1-\delta_2)^\ell$. The reason is that we always force S_2 to accept the first rejection sampling, otherwise B aborts. On interesting point about our attack is that it can be easily be modified to other lattice-based signatures (e.g. [56]) since the signer in such schemes usually outputs only one commitment per session instead of η. The ROS problem in the standard setting is a special case where $\mathcal{S}_{c'} = \mathcal{S}_c = \mathcal{S}$ are finite fields of size q and $\mathcal{X}_\ell = \mathcal{S}^\ell$ [58]. In this setting Schnorr proves that the ℓ-$\mathbf{GROS}$ problem is solvable with probability at most $\binom{Q_H}{\ell+1}/|\mathcal{S}_{c'}| < Q_H^{\ell+1}/q$. Wagner later proposed an algorithm A in the $(\ell := 2^{2\sqrt{\log q}} - 1, Q)$-$\mathbf{GROS}^{\mathsf{A}}_{\mathsf{LHF}}$ experiment with running time $O(2^{2\sqrt{\log q}})$ [60]. The two main reasons that Wagner's algorithm [60] cannot be translated to the ℓ-$\mathbf{GROS}$ problem even with the lattice instantiation from Section 6 are the following. First, let us recall that in Section 6 we select $\mathcal{S} := R_q = \mathbb{Z}_q[X]/\langle X^n + 1\rangle$ to be a cyclotomic ring and $\mathcal{S}_{c'}$ to be a set of short polynomials in R_q. Therefore, we have $\mathcal{S}_{c'} \subsetneq \mathcal{S}_c \subsetneq \mathcal{S}$ and $\mathcal{S}_c, \mathcal{S}_{c'}$ are not finite fields (or even rings). Secondly, compared to the work of Hauck et al. [33], the values s' in a signature have to lie in the set $\mathcal{D}_{s'}$. This imposes a further restriction on the values in the matrix $\mathbf{A}$ and the vector $\boldsymbol{c}$ returned by the $\mathbf{GROS}$ adversary. We believe that the studying this variant of of the $\mathbf{GROS}$ problem further is an interesting problem for future work.

Acknowledgments

We would like to thank Vadim Lyubashevsky for pointing out the flaw in BLAZE and Dominique Schröder for helping us with previous work on blind signatures. We are furthermore very grateful for the anonymous comments by the CRYPTO 2020 reviewers. Eduard Hauck was supported by DFG SPP 1736 Big Data. Eike Kiltz was supported by the BMBF iBlockchain project, the EU H2020 PROMETHEUS project 780701, DFG SPP 1736 Big Data, and the DFG Cluster of Excellence 2092 CASA. Ngoc Khanh Nguyen was supported by the SNSF ERC Transfer Grant CRETP2-166734 FELICITY. Julian Loss was supported by the financial assistance award 70NANB19H126 from U.S. Department of Commerce, National Institute of Standards and Technology.

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. 10, 11
2. M. Abe. A secure three-move blind signature scheme for polynomially many signatures. In B. Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 136–151. Springer, Heidelberg, May 2001. 1, 3
3. M. Abe and T. Okamoto. Provably secure partially blind signatures. In M. Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 271–286. Springer, Heidelberg, Aug. 2000. 22
4. N. A. Alkadri, R. E. Bansarkhani, and J. Buchmann. BLAZE: practical lattice-based blind signatures for privacy-preserving applications. *Financial Cryptography and Data Security - 24rd International Conference, FC 2020*, 2020. 3, 17
5. N. A. Alkadri, R. E. Bansarkhani, and J. Buchmann. On lattice-based interactive protocols with aborts. Cryptology ePrint Archive, Report 2020/007, 2020. `https://eprint.iacr.org/2020/007`. 2, 3, 7, 17
6. M. Backendal, M. Bellare, J. Sorrell, and J. Sun. The fiat-shamir zoo: Relating the security of different signature variants. In N. Gruschka, editor, *Secure IT Systems - 23rd Nordic Conference, NordSec 2018, Oslo, Norway, November 28-30, 2018, Proceedings*, volume 11252 of *Lecture Notes in Computer Science*, pages 154–170. Springer, 2018. 2, 4
7. F. Baldimtsi and A. Lysyanskaya. Anonymous credentials light. In A.-R. Sadeghi, V. D. Gligor, and M. Yung, editors, *ACM CCS 2013*, pages 1087–1098. ACM Press, Nov. 2013. 1
8. 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. 3
9. 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
10. 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. De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 390–399. ACM Press, Oct. / Nov. 2006. 2, 22
11. 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. 11
12. M. Bellare and P. Rogaway. Code-based game-playing proofs and the security of triple encryption. Cryptology ePrint Archive, Report 2004/331, 2004. `http://eprint.iacr.org/2004/331`. 4
13. A. Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In Y. Desmedt, editor, *PKC 2003*, volume 2567 of *LNCS*, pages 31–46. Springer, Heidelberg, Jan. 2003. 3

14. S. Bouaziz-Ermann, S. Canard, G. Eberhart, G. Kaim, A. Roux-Langlois, and J. Traoré. Lattice-based (partially) blind signature without restart. Cryptology ePrint Archive, Report 2020/260, 2020. `https://eprint.iacr.org/2020/260`. 3
15. 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
16. 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
17. 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
18. J. Camenisch, G. Neven, and a. shelat. Simulatable adaptive oblivious transfer. In M. Naor, editor, *EUROCRYPT 2007*, volume 4515 of *LNCS*, pages 573–590. Springer, Heidelberg, May 2007. 7
19. 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
20. 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
21. L. Chen, Y. Cui, X. Tang, D. Hu, and X. Wan. Hierarchical id-based blind signature from lattices. In Y. Wang, Y. Cheung, P. Guo, and Y. Wei, editors, *Seventh International Conference on Computational Intelligence and Security, CIS 2011, Sanya, Hainan, China, December 3-4, 2011*, pages 803–807. IEEE Computer Society, 2011. 3
22. Y. Chen and P. Q. Nguyen. BKZ 2.0: Better lattice security estimates. In D. H. Lee and X. Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 1–20. Springer, Heidelberg, Dec. 2011. 14
23. N. Döttling, N. Fleischhacker, J. Krupp, and D. Schröder. Two-message, oblivious evaluation of cryptographic functionalities. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 619–648. Springer, Heidelberg, Aug. 2016. 1, 3
24. M. Fischlin. Round-optimal composable blind signatures in the common reference string model. In C. Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 60–77. Springer, Heidelberg, Aug. 2006. 1, 2, 3, 7
25. M. Fischlin and D. Schröder. Security of blind signatures under aborts. In S. Jarecki and G. Tsudik, editors, *PKC 2009*, volume 5443 of *LNCS*, pages 297–316. Springer, Heidelberg, Mar. 2009. 7
26. M. Fischlin and D. Schröder. On the impossibility of three-move blind signature schemes. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 197–215. Springer, Heidelberg, May / June 2010. 3
27. G. Fuchsbauer, C. Hanser, and D. Slamanig. Practical round-optimal blind signatures in the standard model. In R. Gennaro and M. J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 233–253. Springer, Heidelberg, Aug. 2015. 3
28. N. Gama and P. Q. Nguyen. Predicting lattice reduction. In N. P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 31–51. Springer, Heidelberg, Apr. 2008. 14
29. W. Gao, Y. Hu, B. Wang, and J. Xie. Identity-based blind signature from lattices in standard model. In K. Chen, D. Lin, and M. Yung, editors, *Information Security and Cryptology - 12th International Conference, Inscrypt 2016, Beijing, China, November 4-6, 2016, Revised Selected Papers*, volume 10143 of *Lecture Notes in Computer Science*, pages 205–218. Springer, 2016. 3
30. W. Gao, Y. Hu, B. Wang, J. Xie, and M. Liu. Identity-based blind signature from lattices. *Wuhan University Journal of Natural Sciences*, 22(4):355–360, 2017. 3
31. S. Garg and D. Gupta. Efficient round optimal blind signatures. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 477–495. Springer, Heidelberg, May 2014. 3
32. S. Garg, V. Rao, A. Sahai, D. Schröder, and D. Unruh. Round optimal blind signatures. In P. Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 630–648. Springer, Heidelberg, Aug. 2011. 3
33. E. Hauck, E. Kiltz, and J. Loss. A modular treatment of blind signatures from identification schemes. In Y. Ishai and V. Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 345–375. Springer, Heidelberg, May 2019. 1, 2, 4, 5, 11, 18, 19, 22, 23, 25, 28
34. A. Juels, M. Luby, and R. Ostrovsky. Security of blind digital signatures (extended abstract). In B. S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 150–164. Springer, Heidelberg, Aug. 1997. 1, 2, 3, 7
35. J. Katz, D. Schröder, and A. Yerukhimovich. Impossibility of blind signatures from one-way permutations. In Y. Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 615–629. Springer, Heidelberg, Mar. 2011. 3
36. A. Langlois and D. Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptogr.*, 75(3):565–599, 2015. 16
37. H. Q. Le, W. Susilo, T. X. Khuc, M. K. Bui, and D. H. Duong. A blind signature from module latices. In *2019 IEEE Conference on Dependable and Secure Computing (DSC)*, pages 1–8. IEEE, 2019. 3
38. V. Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In M. Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 598–616. Springer, Heidelberg, Dec. 2009. 1, 2, 3, 7, 13, 14
39. V. Lyubashevsky. Lattice signatures without trapdoors. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 738–755. Springer, Heidelberg, Apr. 2012. 3

40. V. Lyubashevsky and D. Micciancio. Generalized compact Knapsacks are collision resistant. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *ICALP 2006, Part II*, volume 4052 of *LNCS*, pages 144–155. Springer, Heidelberg, July 2006. 14
41. V. Lyubashevsky and G. Seiler. Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs. In J. B. Nielsen and V. Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 204–224. Springer, Heidelberg, Apr. / May 2018. 16
42. L. Minder and A. Sinclair. The extended k-tree algorithm. In C. Mathieu, editor, *20th SODA*, pages 586–595. ACM-SIAM, Jan. 2009. 17
43. N. K. Nguyen. On the non-existence of short vectors in random module lattices. In S. D. Galbraith and S. Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part II*, volume 11922 of *Lecture Notes in Computer Science*, pages 121–150. Springer, 2019. 16
44. 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
45. T. Okamoto. Efficient blind and partially blind signatures without random oracles. In S. Halevi and T. Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 80–99. Springer, Heidelberg, Mar. 2006. 3
46. 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
47. D. Papachristoudis, D. Hristu-Varsakelis, F. Baldimtsi, and G. Stephanides. Leakage-resilient lattice-based partially blind signatures. Cryptology ePrint Archive, Report 2019/1452, 2019. `https://eprint.iacr.org/2019/1452`. 3
48. C. Peikert and A. Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In S. Halevi and T. Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 145–166. Springer, Heidelberg, Mar. 2006. 14
49. D. Pointcheval. Strengthened security for blind signatures. In K. Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 391–405. Springer, Heidelberg, May / June 1998. 2
50. D. Pointcheval and J. Stern. Provably secure blind signature schemes. In K. Kim and T. Matsumoto, editors, *ASIACRYPT'96*, volume 1163 of *LNCS*, pages 252–265. Springer, Heidelberg, Nov. 1996. 1
51. D. Pointcheval and J. Stern. Security proofs for signature schemes. In U. M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 387–398. Springer, Heidelberg, May 1996. 1, 3
52. D. Pointcheval and J. Stern. New blind signatures equivalent to factorization (extended abstract). In R. Graveman, P. A. Janson, C. Neuman, and L. Gong, editors, *ACM CCS 97*, pages 92–99. ACM Press, Apr. 1997. 1
53. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, June 2000. 1, 2, 3, 22, 23
54. F. Rodriuguez-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
55. P. Rogaway. Formalizing human ignorance. In P. Q. Nguyen, editor, *Progress in Cryptology - VIETCRYPT 06*, volume 4341 of *LNCS*, pages 211–228. Springer, Heidelberg, Sept. 2006. 4
56. M. Rückert. Lattice-based blind signatures. In M. Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 413–430. Springer, Heidelberg, Dec. 2010. 1, 2, 13, 15, 17, 19
57. C.-P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, Jan. 1991. 1
58. 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. 2, 17, 18, 19
59. D. Schröder and D. Unruh. Security of blind signatures revisited. *Journal of Cryptology*, 30(2):470–494, Apr. 2017. 7
60. D. Wagner. A generalized birthday problem. In M. Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 288–303. Springer, Heidelberg, Aug. 2002. 2, 17, 19
61. X. Yi, K.-Y. Lam, and D. Gollmann. A new blind ECDSA scheme for bitcoin transaction anonymity. Cryptology ePrint Archive, Report 2018/660, 2018. `https://eprint.iacr.org/2018/660`. 1
62. L. Zhang and Y. Ma. A lattice-based identity-based proxy blind signature scheme in the standard model. *Mathematical Problems in Engineering*, 2014, 2014. 3
63. H. Zhu, Y. Tan, X. Zhang, L. Zhu, C. Zhang, and J. Zheng. A round-optimal lattice-based blind signature scheme for cloud services. *Future Generation Comp. Syst.*, 73:106–114, 2017. 3

A Proof of Theorem 4

In this section, we present the proof of Theorem 4. We begin by introducing two crucial lemmas for the proof and then give a proof intuition.

A.1 Prerequisites

THE SUBSET FORKING LEMMA. We recall the Subset Forking Lemma, which was introduced in [33] as a generalization of the General Forking Lemma of Bellare and Neven [10].

Lemma 8 (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 $0 \leq j \leq Q$ and $\sigma \in \Sigma$. We partition its input space $\mathcal{I} \times \Omega \times \mathcal{H}^Q$ into sets $\mathcal{W}_1, \dots, \mathcal{W}_Q$ 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

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

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

Then

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

GENERALIZED SPLITTING LEMMA. We begin by proving a simple generalization of the well known splitting lemma [53]. This lemma is also sometimes referred to as the 'heavy row' lemma in the literature (e.g. [3]).

Lemma 9 (Generalized Splitting Lemma). *Let $n \in \mathbb{N}$ and $\mathcal{X}_1, ..., \mathcal{X}_n$ be sets of finite size. Let $\mathcal{B} \subset \mathcal{X}_1 \times \cdots \times \mathcal{X}_n := \boldsymbol{\mathcal{X}}$ be such that*

$$\Pr_{\boldsymbol{x} \xleftarrow{\$} \boldsymbol{\mathcal{X}}} [\boldsymbol{x} \in \mathcal{B}] := \varepsilon.$$

For any $\mathcal{S} \subset [n], \alpha \leq \varepsilon$ and $i_1 < \cdots < i_{|\bar{\mathcal{S}}|}$ and $\boldsymbol{\mathcal{X}}'_{\mathcal{S}} := \mathcal{X}_{i_1} \times \cdots \times \mathcal{X}_{i_{|\bar{\mathcal{S}}|}}$, define

$$\mathcal{B}_{\mathcal{S},\alpha} := \big\{ (x_1, ..., x_n) \in \boldsymbol{\mathcal{X}} \mid \Pr_{\boldsymbol{x}' \xleftarrow{\$} \boldsymbol{\mathcal{X}}'_{\mathcal{S}}} [\boldsymbol{x}_{\mathcal{S},\boldsymbol{x}'} \in \mathcal{B}] \geq \varepsilon - \alpha \big\},$$

where

$$\boldsymbol{x}_{\mathcal{S},\boldsymbol{x}'} := \begin{cases} \boldsymbol{x}_i, \text{ if } i \in \mathcal{S}, \\ \boldsymbol{x}'_i, \text{ otherwise} \end{cases}.$$

Then the following statements hold:

(i) $\Pr_{\boldsymbol{x} \xleftarrow{\$} \boldsymbol{\mathcal{X}}} [\boldsymbol{x} \in \mathcal{B}_{\mathcal{S},\alpha}] \geq \alpha$

(ii) $\forall \boldsymbol{x} \in \mathcal{B}_{\mathcal{S},\alpha} \colon \Pr_{\boldsymbol{x}' \xleftarrow{\$} \boldsymbol{\mathcal{X}}'_{\mathcal{S}}} [\boldsymbol{x}_{\mathcal{S},\boldsymbol{x}'} \in \mathcal{B}] \geq \varepsilon - \alpha$

(iii) $\Pr_{\boldsymbol{x} \xleftarrow{\$} \boldsymbol{\mathcal{X}}} [\boldsymbol{x} \in \mathcal{B}_{\mathcal{S},\alpha} \mid \boldsymbol{x} \in \mathcal{B}] \geq \alpha / \varepsilon$

Proof. We argue along the lines of [53]. Item (ii) follows directly from the definition of $\mathcal{B}_{\mathcal{S},\alpha}$. To see that Item (i) holds, suppose, toward a contradiction, that $\Pr_{\boldsymbol{x} \xleftarrow{\$} \boldsymbol{\mathcal{X}}} [\boldsymbol{x} \in \mathcal{B}_{\mathcal{S},\alpha}] < \alpha$. By law of total probability, we have

$$\begin{aligned} \varepsilon &= \Pr_{\boldsymbol{x} \xleftarrow{\$} \boldsymbol{\mathcal{X}}} [\boldsymbol{x} \in \mathcal{B}] \\ &= \Pr_{\boldsymbol{x} \xleftarrow{\$} \boldsymbol{\mathcal{X}}} [\boldsymbol{x} \in \mathcal{B}_{\mathcal{S},\alpha}] \cdot \Pr_{\boldsymbol{x} \xleftarrow{\$} \boldsymbol{\mathcal{X}}} [\boldsymbol{x} \in \mathcal{B} \mid \boldsymbol{x} \in \mathcal{B}_{\mathcal{S},\alpha}] + \Pr_{\boldsymbol{x} \xleftarrow{\$} \boldsymbol{\mathcal{X}}} [\boldsymbol{x} \in \bar{\mathcal{B}}_{\mathcal{S},\alpha}] \cdot \Pr_{\boldsymbol{x} \xleftarrow{\$} \boldsymbol{\mathcal{X}}} [\boldsymbol{x} \in \mathcal{B} \mid \boldsymbol{x} \in \bar{\mathcal{B}}_{\mathcal{S},\alpha}] \\ &< \alpha \cdot 1 + 1 \cdot (\varepsilon - \alpha) = \varepsilon, \end{aligned}$$

a contradiction. To see Item (iii), we consider

$$\begin{aligned}
&\Pr_{\boldsymbol{x} \xleftarrow{\$} \boldsymbol{\mathcal{X}}}[\boldsymbol{x} \in \mathcal{B}_{\mathcal{S},\alpha} \mid \boldsymbol{x} \in \mathcal{B}] \\
&= 1 - \Pr_{\boldsymbol{x} \xleftarrow{\$} \boldsymbol{\mathcal{X}}}[\boldsymbol{x} \in \bar{\mathcal{B}}_{\mathcal{S},\alpha} \mid \boldsymbol{x} \in \mathcal{B}] \\
&= 1 - \Pr[\boldsymbol{x} \in \mathcal{B} \mid \boldsymbol{x} \in \bar{\mathcal{B}}_{\mathcal{S},\alpha}] \cdot \frac{\Pr[\boldsymbol{x} \in \bar{\mathcal{B}}_{\mathcal{S},\alpha}]}{\Pr[\boldsymbol{x} \in \mathcal{B}]} \\
&\geq 1 - \Pr[\boldsymbol{x} \in \mathcal{B} \mid \boldsymbol{x} \in \bar{\mathcal{B}}_{\mathcal{S},\alpha}] \cdot \frac{1}{\Pr[\boldsymbol{x} \in \mathcal{B}]} = 1 - \frac{\varepsilon - \alpha}{\varepsilon} = \alpha/\varepsilon.
\end{aligned}$$

A.2 Proof Intuition and Differences to [33]

The proof of Theorem 4 follows the same arguments as the proof in [33], and so we focus here on highlighting the differences to their proof. At the heart of the proof in both [53] and [33] lies the observation that the same transcript of the protocol execution between the adversary and the simulator could have resulted from distinct secret keys sk_1 and sk_2, both of which map to the same public key pk under F. The proof then leverages a subtle argument by the probabilistic method to ensure that the adversary cannot force certain variables in the view of the simulator that depend on the actual choice of sk_1 or sk_2 to always take the same value, with overwhelming probability, when the adversary is run with two distinct sets of challenges $\boldsymbol{h}$ and $\boldsymbol{h}$'. In their proof, this corresponds to the variable $\hat{\chi}$ which depends on both the secret key as well as the vector $\boldsymbol{h}$ of challenge values. To analyze these probabilities, the proofs of [53] and [33] follow the usual paradigm of derandomizing the adversary, by viewing it as a deterministic algorithm that takes as input a tuple consisting of an instance in the form of a secret key sk, a selection of randomness required in the experiment, and a fixed set of challenges, $\boldsymbol{h}$. On top of this, their proofs then exploit the fact that a public key pk has multiple preimages under F. This is used in the proof by defining a mapping Φ that maps any tuple from the adversary's underlying input space to another input tuple, which will result in the same view for the adversary. Subsequently, the proofs of [53] and [33] then argue over particular properties of Φ, one of which is that it defines a bijective mapping on the set of such input tuples that lead the adversary to a successful run. The difficulty that occurs in our context is that (because of imperfect correctness), while we can still define the mapping Φ in the same way as [33], it is possible that Φ maps valid tuples to invalid ones, which would not occur in an honest execution of the simulation. This unravels the subtle argument in [53,33]. Thus, an additional difficulty of our proof is to show that, for a not too small fraction of such tuples, their image under Φ still results in a valid tuple that could occur as the result of an actual execution of the experiment. Here, we can once again rely on the generalized splitting lemma, introduced in the previous section which we can use to force a certain fraction of 'good tuples' to exist (within the set of all tuples).

A.3 The Reduction Algorithm

Let M be an adversary against $(\varepsilon, t, Q_{\mathsf{V}}, Q_{\mathsf{P}_1}, Q_{\mathsf{P}_2})$-**OMMIM** of $\mathsf{ID}[\mathsf{LHF}]$ relative to par. We show how to construct an adversary B against (ε', t')-**CR** of LHF relative to par. The first part of our proof follows very closely along the lines of [33]. Without loss of generality, we will assume 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 $1 \leq i \leq Q_{\mathsf{P}_2} + 1$, the idea is to define wrapper algorithms A_i which 'sandbox' M in such a way that A_i is deterministic and self-contained, i.e., has a simple input-output behaviour (as opposed to M, who can ask signing and challenge queries). A_i (for some i) is then later invoked by adversary B to break collision resistance of LHF. 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 $\mathtt{Ch}$. Throughout the proof, we will denote with $|\mathcal{C}| \geq 2^{2\kappa}$ the size of the challenge space $\mathcal{C} = \mathcal{C}(par) = \mathcal{S}_{c'}$. 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 the same notation as [33]. Variables $\hat{\boldsymbol{J}}_i, \hat{\boldsymbol{\chi}}_i, \hat{\boldsymbol{s}}'$, and $\hat{\boldsymbol{h}}_i$ are defined on Lines 35 through 38 of Figure 11 and are introduced to simplify the referencing of values associated with successful calls to oracle $\mathtt{V}_2(vSid, \cdot)$. 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.

As A_i is deterministic, an execution of A_i is fixed 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 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,

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

In this way, $\hat{\boldsymbol{J}}_i, \hat{\boldsymbol{\chi}}_i, \hat{\boldsymbol{s}}'$, and $\hat{\boldsymbol{h}}_i$ can be seen as random variables whose randomness is taken from the uniform distribution on $(\mathcal{I} \times \Omega \times \mathcal{C}^{Q_{\mathsf{V}}})$. Since their outcome is uniquely determined given $(I, \omega, \boldsymbol{h}) \in \mathcal{W}$, we write

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

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

```
Adversary $\mathsf{A}_i(I = sk, \boldsymbol{h}; \omega)$:
Parse $(\omega_{\mathsf{M}}, \boldsymbol{r}_{1,1}, \ldots, \boldsymbol{r}_{Q_{\mathsf{P}_1},\eta}) \leftarrow \omega$
$\boldsymbol{R}_{i,j} \leftarrow \mathsf{F}(\boldsymbol{r}_{i,j})$
$pk \leftarrow \mathsf{F}(sk)$
$ctr \leftarrow 0; pSid \leftarrow 0; vSid \leftarrow 0$
$\mathsf{M}^{\mathtt{P_1},\mathtt{P_2},\mathtt{Ch},\mathtt{V_2}}(pk)$
$\ell(\mathsf{M}) \leftarrow \#\{k \mid \mathbf{vSess}_k = \mathtt{closed} \wedge \boldsymbol{b}'_k = 1\}$
$Q_{\mathsf{P}_2}(\mathsf{M}) \leftarrow \#\{k \mid \mathbf{pSess}_k = \mathtt{closed}\}$
$Q_{\mathsf{P}_1}(\mathsf{M}) \leftarrow \#\{k \mid \mathbf{pSess}_k = \mathtt{open}\}$
$Q_{\mathsf{V}}(\mathsf{M}) \leftarrow vSid$
If $(\ell(\mathsf{M}) \geq Q_{\mathsf{P}_2}(\mathsf{M}) + 1)$: Return $(\hat{\boldsymbol{J}}_i, \hat{\boldsymbol{\chi}}_i)$
Return $(\hat{\boldsymbol{J}}_i, \hat{\boldsymbol{\chi}}_i) \leftarrow (0, 0)$

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

Procedure $\mathsf{P}_2(pSid, c)$
If $\mathbf{pSess}_{pSid} \neq \mathtt{open}$: Return $\perp$
$\mathbf{pSess}_{pSid} \leftarrow \mathtt{closed}$
If $c \notin \mathcal{S}_c$: Return $\perp$
$\boldsymbol{c}_{pSid} \leftarrow c$
For $i \in [\eta]$:
    $\boldsymbol{s}_i \leftarrow c \cdot sk + \boldsymbol{r}_{pSid,i}$
    If $\boldsymbol{s}_i \in \mathcal{D}_s$
        Return $\boldsymbol{s}_{pSid}$
Return $\perp$

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

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

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

$$\Pr_{\boldsymbol{h'} \xleftarrow{\$} \mathcal{C}^{Q_{\mathsf{V}}} \mid \boldsymbol{h}_{[j-1]}} [\hat{\boldsymbol{J}}_i(I, \omega, \boldsymbol{h'}) = j \wedge \hat{\boldsymbol{\chi}}_i(I, \omega, \boldsymbol{h'}) = c], \tag{9}$$

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

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

as the lexicographically first value c s.t. the probability in (9) is maximized when $(I, \omega, \boldsymbol{h}), j, i$ are fixed. To simplify notations, we also introduce $C_i(I, \omega, \boldsymbol{h}) = c_{i, \hat{\boldsymbol{J}}_i(I,\omega,\boldsymbol{h})}(I, \omega, \boldsymbol{h})$.

```
Adversary B:
$i^* \xleftarrow{\$} [Q_{\mathsf{P}_2}+1]$
$\boldsymbol{h} \xleftarrow{\$} \mathcal{C}^{Q_\mathsf{V}}$
$\omega \xleftarrow{\$} \Omega$
$sk \xleftarrow{\$} \mathcal{D}_{sk}$
$(\hat{\boldsymbol{J}}_{i^*}, \hat{\boldsymbol{\chi}}_{i^*}) \leftarrow \mathsf{A}_{i^*}(I = sk, \boldsymbol{h}; \omega)$          //First execution of $\mathsf{A}_{i^*}$
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 $\mathsf{A}_{i^*}$
If $(\hat{\boldsymbol{J}}'_{i^*} = \hat{\boldsymbol{J}}_{i^*}) \wedge (\hat{\boldsymbol{\chi}}_{i^*} \neq \hat{\boldsymbol{\chi}}'_{i^*})$: Return $(\hat{\boldsymbol{\chi}}_{i^*}, \hat{\boldsymbol{\chi}}'_{i^*})$
Return $\perp$
```

Fig. 12. Adversary B against collision resistance of LHF relative to fixed par.

For fixed i, j, we can now define $\mathcal{B}_{i,j} \subset \mathcal{W}$ as

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

and

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

We reuse Lemma 7.1 from [33], which follows in exactly the same way as in their proof. We restate their proof for completeness in Section A.4.

Lemma 10. *For all i, j: $\delta_{i,j} \geq \beta_{i,j} \left(\frac{\beta_{i,j}}{8} - \frac{1}{2|\mathcal{C}|} \right)$.*

Lemma 11. *There exist $i \in [Q_{\mathsf{P}_2} + 1], j \in [Q_\mathsf{V}]$ such that $\beta_{i,j} > \left(\frac{\varepsilon^2}{8} - \frac{Q_\mathsf{V}^{Q_{\mathsf{P}_2}+1} \cdot \binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{\left(\frac{|\mathcal{C}|}{|\mathcal{S}_c|}\right)^{Q_{\mathsf{P}_2}} \cdot |\mathcal{C}|} \right) \cdot \frac{1}{2Q_\mathsf{V}(Q_{\mathsf{P}_2}+1)}$.*

The proof of this lemma is postponed to Section A.5.

Adversary B against Collision Resistance of LHF. We next describe the adversary B depicted in Figure 12, against the collision resistance of LHF. B first samples randomness $\omega \xleftarrow{\$} \Omega$, a secret key $sk \xleftarrow{\$} \mathcal{D}_{sk}$, 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 $(I = sk, \boldsymbol{h}; \omega)$. To run A_{i^*} a second time, it then 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^*} again, this time with the same randomness ω and the same instance I, but with $\boldsymbol{h'}$ instead of $\boldsymbol{h}$. Note that by definition of A_{i^*},

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

whenever B does not abort. Note that A_{i^*} sees identical answers for the first $\hat{\boldsymbol{J}}_{i^*} - 1$ queries to $\mathtt{Ch}$, and so it behaves identically in both runs up to the point where it receives the answer to the $\hat{\boldsymbol{J}}_{i^*}$-th query to $\mathtt{Ch}$. In particular, A_{i^*} poses the same $\hat{\boldsymbol{J}}_{i^*}$-th query to $\mathtt{Ch}$ 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

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

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 10. By Lemma 11 we obtain the final bound

$$\begin{aligned}
\varepsilon' &\geq \frac{\frac{\varepsilon^2}{8} - \frac{Q_{\mathsf{V}}^{Q_{\mathsf{P}_2}+1} \cdot \binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{\left(\frac{|\mathcal{C}|}{|\mathcal{S}_c|}\right)^{Q_{\mathsf{P}_2}} \cdot |\mathcal{C}|}}{32 Q_{\mathsf{V}}^2 (Q_{\mathsf{P}_2} + 1)^3} \cdot \left(\frac{\varepsilon^2}{8} - \frac{Q_{\mathsf{V}}^{Q_{\mathsf{P}_2}+1} \cdot \binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{\left(\frac{|\mathcal{C}|}{|\mathcal{S}_c|}\right)^{Q_{\mathsf{P}_2}} \cdot |\mathcal{C}|} - \frac{16 Q_{\mathsf{V}}^2 (Q_{\mathsf{P}_2} + 1)^2}{|\mathcal{C}|} \right) \\
&= O\left(\left(\varepsilon^2 - \frac{(Q_{\mathsf{V}} Q_{\mathsf{P}_1})^{Q_{\mathsf{P}_2}+1}}{\left(\frac{|\mathcal{C}|}{|\mathcal{S}_c|}\right)^{Q_{\mathsf{P}_2}} \cdot |\mathcal{C}|} \right)^2 \frac{1}{Q_{\mathsf{V}}^2 Q_{\mathsf{P}_2}^3} \right),
\end{aligned}$$

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

A.4 Proof of Lemma 10

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

$$\begin{aligned}
\alpha_{i,j}(I, \omega, \boldsymbol{h}, d) := &\Pr_{\boldsymbol{h'} \xleftarrow{\$} \mathcal{C}^{Q_{\mathsf{V}}} | \boldsymbol{h}_{[j-1]}} [\hat{\boldsymbol{\chi}}_i(I, \omega, \boldsymbol{h'}) \neq d \wedge \hat{\boldsymbol{J}}_i(I, \omega, \boldsymbol{h'}) = j] \\
&\geq \mu_{i,j}(I, \omega, \boldsymbol{h})/2,
\end{aligned} \tag{10}$$

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, we define the boolean variable $B(s)$ as 1 if s is true and 0 otherwise. Now notice that (10) implies the theorem statement since

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

In the last step, we have applied linearity and monotonicity of the expectation as well as (10).

$$
\begin{aligned}
&\frac{1}{2}\sum_{d}\mathbf{E}_{I,\omega,\boldsymbol{h}}[B(\hat{\boldsymbol{\chi}}_i(I,\omega,\boldsymbol{h})=d\wedge\hat{\boldsymbol{J}}_i(I,\omega,\boldsymbol{h})=j)\cdot\mu_{i,j}(I,\omega,\boldsymbol{h})]\\
&=\frac{1}{2}\cdot\sum_{d}\Pr_{(I,\omega,\boldsymbol{h})\xleftarrow{\$}(\mathcal{I}\times\Omega\times\mathcal{C}^{Q_V}),\boldsymbol{h'}\xleftarrow{\$}\mathcal{C}^{Q_V}|\boldsymbol{h}_{[j-1]}}\begin{bmatrix}\hat{\boldsymbol{\chi}}_i(I,\omega,\boldsymbol{h})=d\wedge\hat{\boldsymbol{J}}_i(I,\omega,\boldsymbol{h})=j\\ \wedge(I,\omega,\boldsymbol{h'})\in\mathcal{B}_{i,j}\wedge\boldsymbol{h}_j\neq\boldsymbol{h}'_j\end{bmatrix}\\
&=\frac{1}{2}\cdot\Pr_{(I,\omega,\boldsymbol{h})\xleftarrow{\$}(\mathcal{I}\times\Omega\times\mathcal{C}^{Q_V}),\boldsymbol{h'}\xleftarrow{\$}\mathcal{C}^{Q_V}|\boldsymbol{h}_{[j-1]}}\begin{bmatrix}\hat{\boldsymbol{J}}_i(I,\omega,\boldsymbol{h})=j\\ \wedge(I,\omega,\boldsymbol{h'})\in\mathcal{B}_{i,j}\wedge\boldsymbol{h}_j\neq\boldsymbol{h}'_j\end{bmatrix} &(11)\\
&\geq\frac{1}{2}\cdot\Pr_{(I,\omega,\boldsymbol{h})\xleftarrow{\$}(\mathcal{I}\times\Omega\times\mathcal{C}^{Q_V}),\boldsymbol{h'}\xleftarrow{\$}\mathcal{C}^{Q_V}|\boldsymbol{h}_{[j-1]}}[(I,\omega,\boldsymbol{h})\in\mathcal{B}_{i,j}\wedge(I,\omega,\boldsymbol{h'})\in\mathcal{B}_{i,j}\wedge\boldsymbol{h}_j\neq\boldsymbol{h}'_j] &(12)\\
&=\frac{1}{2}\cdot\mathtt{frk}(\mathcal{B}_{i,j},j) &(13)\\
&\geq\beta_{i,j}\left(\beta_{i,j}/8-\frac{1}{2|\mathcal{C}|}\right). &(14)
\end{aligned}
$$

Fom (11) to (12), 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 step from (13) to (14) follows from Lemma 8. We prove (10) by analyzing two cases. For all $I,\omega,\boldsymbol{h},d$, we define

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

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 $d\neq c_{i,j}(I,\omega,\boldsymbol{h})$. This is because if $d=c_{i,j}(I,\omega,\boldsymbol{h})$, then

$$
\begin{aligned}
\gamma_{i,j}(I,\omega,\boldsymbol{h},d)&\leq\Pr[\hat{\boldsymbol{\chi}}_i(I,\omega,\boldsymbol{h'})=c_{i,j}(I,\omega,\boldsymbol{h})\wedge(I,\omega,\boldsymbol{h'})\in\mathcal{B}_{i,j}]\\
&=\Pr[\hat{\boldsymbol{\chi}}_i(I,\omega,\boldsymbol{h'})=c_{i,j}(I,\omega,\boldsymbol{h'})\wedge(I,\omega,\boldsymbol{h'})\in\mathcal{B}_{i,j}]\\
&=\Pr[\hat{\boldsymbol{\chi}}_i(I,\omega,\boldsymbol{h'})=C_i(I,\omega,\boldsymbol{h'})\wedge(I,\omega,\boldsymbol{h'})\in\mathcal{B}_{i,j}]=0.
\end{aligned}
$$

For the first equality, we used the fact that $c_{i,j}(I,\omega,\boldsymbol{h})=c_{i,j}(I,\omega,\boldsymbol{h'})$ for any $\boldsymbol{h}$ and $\boldsymbol{h}'$ which have the same $j-1$ first entries. For the second equality, we have again used the fact that $(I,\omega,\boldsymbol{h'})\in\mathcal{B}_{i,j}$ implies $\hat{\boldsymbol{J}}_i(I,\omega,\boldsymbol{h'})=j$. This trivializes the claim and so we assume in the following that $d\neq c_{i,j}(I,\omega,\boldsymbol{h})$. We now continue with

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

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,

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

This proves (10) and hence the lemma.

A.5 Proof of Lemma 11

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 19 of Figure 11.

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

Lemma 12. *Consider the mapping*

$$\begin{aligned}
&\varPhi : \mathcal{W} \longrightarrow (\mathcal{I} \times \varOmega \times \mathcal{C}^{Q_{\mathsf{V}}}) \\
&\big(sk, (\omega_{\mathsf{M}}, \boldsymbol{r}_{1,1}, \ldots, \boldsymbol{r}_{Q_{\mathsf{P}_2}, \eta}), \boldsymbol{h}\big) \mapsto \\
&\big(sk + z^*, (\omega_{\mathsf{M}}, \boldsymbol{r}_{1,1} - z^* \cdot \boldsymbol{c}(I, \omega, \boldsymbol{h}), \ldots, \boldsymbol{r}_{Q_{\mathsf{P}_1}, \eta} - z^* \cdot \boldsymbol{c}(I, \omega, \boldsymbol{h})), \boldsymbol{h}\big)\,,
\end{aligned}$$

where we make the convention that for $v \in \mathcal{D} \cup \mathcal{C} \cup \mathcal{R}, v \cdot \perp := 0$. Let $\mathcal{W}_{inj} \subset \mathcal{W}$ be the set of $(I, \omega, \boldsymbol{h})$ s.t. $\varPhi(I, \omega, \boldsymbol{h}) \in \varPhi(\mathcal{W}) \cap \mathcal{W}$. Then the restriction $\varPhi_{inj} : \mathcal{W}_{inj} \longrightarrow \varPhi(\mathcal{W}) \cap \mathcal{W}$ of $\varPhi$ to $\mathcal{W}_{inj}$ is injective.

For the proof we require the following claim.

Claim. Let $(I, \omega, \boldsymbol{h}) \in \mathcal{W}$. If $\varPhi(I, \omega, \boldsymbol{h}) \in \mathcal{W}$, then the tuples $(I, \omega, \boldsymbol{h})$ and $\varPhi(I, \omega, \boldsymbol{h})$ fix the same execution of M.

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

- **Initial input to** M**.** Since $\varPhi$ does not alter the values in 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 02 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** $\mathtt{V}_2$**.** Oracle $\mathtt{V}_2$ 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}.
- **Outputs of oracle** $\mathtt{P}_2$**.** Oracle $\mathtt{P}_2$ consecutively returns the values from $\boldsymbol{s} = \boldsymbol{c} \cdot sk + \boldsymbol{r}$, as defined in Line 21 of Figure 11 (or $\perp$, in case $\boldsymbol{s} \notin \mathcal{D}_s$). Note that the first value $\boldsymbol{c}_1$ in both executions is the same (as it only depends on values that we have already argued to remain the same in both executions), i.e., $\boldsymbol{c}_1 = \boldsymbol{c}_1(I, \omega, \boldsymbol{h}) = \boldsymbol{c}_1(\varPhi(I, \omega, \boldsymbol{h}))$. Thus, $\boldsymbol{s}_1(I, \omega, \boldsymbol{h}) = \boldsymbol{r}_1 + sk \cdot \boldsymbol{c}_1(I, \omega, \boldsymbol{h}) = \boldsymbol{r}_1 - z^* \cdot \boldsymbol{c}_1(I, \omega, \boldsymbol{h}) + z^* \cdot \boldsymbol{c}_1(I, \omega, \boldsymbol{h}) + sk \cdot \boldsymbol{c}_1(I, \omega, \boldsymbol{h}) = (\boldsymbol{r}_1 - z^* \cdot \boldsymbol{c}_1(\varPhi(I, \omega, \boldsymbol{h}))) + (sk + z^*) \cdot \boldsymbol{c}_1(\varPhi(I, \omega, \boldsymbol{h})) = \boldsymbol{s}_1(\varPhi(I, \omega, \boldsymbol{h}))$, where in the second to last step, we have used the distributive law over the module formed by $\mathcal{C}$ and $\mathcal{D}$. (Here, we overload the notation of $\boldsymbol{c}_1, \boldsymbol{r}_1$ to denote the values corresponding to the first session in which these values are not $\perp$). By a simple inductive argument, it now follows that $\boldsymbol{s}(I, \omega, \boldsymbol{h}) = \boldsymbol{s}(\varPhi(I, \omega, \boldsymbol{h}))$.

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

Proof (Proof of Lemma 12). Suppose $\varPhi_{inj}$ is not injective. Thus, for distinct tuples $(I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) \neq (I', (\omega'_{\mathsf{M}}, \boldsymbol{r'}), \boldsymbol{h'})$ in $\mathcal{W}_{inj}$, we have $\varPhi_{inj}\,(I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) = \varPhi_{inj}(I', (\omega'_{\mathsf{M}}, \boldsymbol{r'}), \boldsymbol{h'})$. 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 $\varPhi_{inj}\,(I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) = \varPhi_{inj}(I', (\omega'_{\mathsf{M}}, \boldsymbol{r'}), \boldsymbol{h'})$, by Claim A.5, $(I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h})$ 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 $(I, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) = (I', (\omega'_{\mathsf{M}}, \boldsymbol{r'}), \boldsymbol{h'})$.

We now introduce the following notation. Let $\mathcal{B} = \bigcup\limits_{i,j} \mathcal{B}_{i,j}$ 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 lemma from [33] can be proven in an almost verbatim manner for our setting, so we do not reprove it here.

Lemma 13. *For any* $(\omega, I) = ((\omega_{\mathsf{M}}, \boldsymbol{r}), I) \in \varOmega \times \mathcal{I}$,

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

Proof. Toward a contradiction, assume that for some $\varOmega \times \mathcal{I}$

$$\Pr_{\boldsymbol{h} \xleftarrow{\$} \mathcal{C}^{Q_\mathsf{V}}} \begin{bmatrix} (I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{h}) \in \mathcal{G} \\ \wedge \varPhi\,(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{h}) \in \mathcal{G} \end{bmatrix} > \frac{Q_\mathsf{V}^{Q_{\mathsf{P}_2}+1} \cdot \binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{\left(\frac{|\mathcal{C}|}{|\mathcal{S}_c|}\right)^{Q_{\mathsf{P}_2}} \cdot |\mathcal{C}|} \,. \tag{15}$$

Thus, there exist a set $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{Q_{\mathsf{P}_2}+1}\}$ of $Q_{\mathsf{P}_2} + 1$ distinct indices from $[Q_\mathsf{V}]$, such that

$$\Pr_{\boldsymbol{h} \xleftarrow{\$} \mathcal{C}^{Q_\mathsf{V}}} \begin{bmatrix} (I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{h}) \in \mathcal{G} \wedge\ \varPhi\,(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{h}) \in \mathcal{G} \\ \wedge \forall i : \hat{\boldsymbol{J}}_i\,(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{h}) = \boldsymbol{u}_i \end{bmatrix} > \frac{\binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{\left(\frac{|\mathcal{C}|}{|\mathcal{S}_c|}\right)^{Q_{\mathsf{P}_2}} \cdot |\mathcal{C}|} \,.$$

Next, consider a vector $\boldsymbol{d} \in \{\mathcal{S}_c \cup \{\bot\}\}^{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}$ of the following format. In any position i corresponding to one the Q_{P_1} abandoned sessions, $\boldsymbol{d}_i = \bot$. The remaining Q_{P_2} positions are in the range $\mathcal{S}_c$. Since there are $\binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}} \cdot |\mathcal{S}_c|^{Q_{\mathsf{P}_2}}$ such vectors, there must exist one vector $\boldsymbol{d}$ that satisfies

$$\begin{aligned} &\Pr_{\boldsymbol{h} \xleftarrow{\$} \mathcal{C}^{Q_\mathsf{V}}} \begin{bmatrix} (I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{h}) \in \mathcal{G} \wedge\ \varPhi\,(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{h}) \in \mathcal{G} \\ \wedge \forall i : \hat{\boldsymbol{J}}_i\,(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{h}) = \boldsymbol{u}_i \wedge \boldsymbol{c}\,(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{h}) = \boldsymbol{d} \end{bmatrix} \\ &> \frac{1}{\left(\frac{|\mathcal{C}|}{|\mathcal{S}_c|}\right)^{Q_{\mathsf{P}_2}} \cdot |\mathcal{C}|\, |\mathcal{S}_c|^{Q_{\mathsf{P}_2}}} = \frac{1}{|\mathcal{C}|^{Q_{\mathsf{P}_2}+1}} \,. \end{aligned}$$

Finally, there exists a set $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{Q_\mathsf{V}} - Q_{\mathsf{P}_2} - 1\}$ of $Q_\mathsf{V} - Q_{\mathsf{P}_2} - 1$ distinct indices from $[Q_\mathsf{V}] \setminus \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{Q_{\mathsf{P}_2}+1}\}$ and a vector $(\tilde{h}_1, \ldots, \tilde{h}_{Q_\mathsf{V}-Q_{\mathsf{P}_2}-1}) \in \mathcal{C}^{Q_\mathsf{V}-Q_{\mathsf{P}_2}-1}$, such that:

$$\begin{aligned} &\Pr_{\boldsymbol{h} \xleftarrow{\$} \mathcal{C}^{Q_\mathsf{V}}} \begin{bmatrix} (I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{h}) \in \mathcal{G} \wedge\ \varPhi\,(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{h}) \in \mathcal{G} \wedge \forall i : \hat{\boldsymbol{J}}_i(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{h'}) = \boldsymbol{u}_i \\ \wedge \boldsymbol{c}\,(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{h}) = \boldsymbol{d}\, \forall m : \boldsymbol{h}_{\boldsymbol{x}_m} = \tilde{\boldsymbol{h}}_{\boldsymbol{x}_m} \end{bmatrix} \\ &> \frac{1}{|\mathcal{C}|^{Q_{\mathsf{P}_2}+1}\, |\mathcal{C}|^{Q_\mathsf{V}-Q_{\mathsf{P}_2}-1}} = \frac{1}{|\mathcal{C}|^{Q_\mathsf{V}}} \,. \end{aligned} \tag{16}$$

Hence, there exist at least two vectors $\boldsymbol{k}, \boldsymbol{k'} \in \mathcal{C}^{Q_{Q_\mathsf{V}}}$ such that the condition inside the probability term 16 is true. Denote these vectors as $\boldsymbol{k} \neq \boldsymbol{k'}$. Now there exists an index $i \in \mathcal{V}$ s.t. $\boldsymbol{k}_i \neq \boldsymbol{k}'_i$. W.l.o.g., let i be the smallest such index. This implies that $\forall j < i : \boldsymbol{k}_j = \boldsymbol{k}'_j$ and $\boldsymbol{k}_i \neq \boldsymbol{k}'_i$. Moreover, we know that $i \in \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{Q_{\mathsf{S}_2}+1}\}$. Let $i = \boldsymbol{u}_j, j \in [Q_{\mathsf{P}_2} + 1]$. The equality $\boldsymbol{k}|_i = \boldsymbol{k}'|_i$ implies that

$$\begin{aligned} C_j(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k}) &= c_{j, \hat{\boldsymbol{J}}j(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k})} \\ &= c_{j,i}(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k}) = c_{j,i}(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k'}) \\ &= c_{j, \hat{\boldsymbol{J}}j(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k'})} = C_j(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k'}). \end{aligned} \tag{17}$$

From Lemma A.5 we know that $\hat{\boldsymbol{J}}j(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k}) = \hat{\boldsymbol{J}}j(\varPhi(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k})) = i$ and that $\boldsymbol{s}_j(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k}) = \boldsymbol{s}_j(\varPhi(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k}))$. We also know that $(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k}) \in \mathcal{G}$ and $\varPhi(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k}) \in \mathcal{G}$. This implies that $\boldsymbol{\chi}_j(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k}) = \boldsymbol{C}_j(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k}) = \boldsymbol{s}_j(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k}) - sk \cdot \boldsymbol{k}_i$ and $\boldsymbol{\chi}_j(\varPhi(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k})) = \boldsymbol{C}_j(\varPhi(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k})) = \boldsymbol{s}_j(\varPhi(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k})) - (sk + z^*) \cdot \boldsymbol{k}_i$. So we infer

$$\begin{aligned} \boldsymbol{C}_j(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k}) &= \boldsymbol{s}_j(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k}) - sk \cdot \boldsymbol{k}_i \\ &= \boldsymbol{s}_j(\varPhi(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k})) - sk \cdot \boldsymbol{k}_i \\ &= \boldsymbol{s}_j(\varPhi(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k})) - sk \cdot \boldsymbol{k}_i + z^* \cdot \boldsymbol{k}_i - z^* \cdot \boldsymbol{k}_i \\ &= \boldsymbol{s}_j(\varPhi(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k})) - (sk + z^*) \cdot \boldsymbol{k}_i + z^* \cdot \boldsymbol{k}_i \\ &= \boldsymbol{C}_j(\varPhi(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k})) + z^* \cdot \boldsymbol{k}_i \\ &= \boldsymbol{C}_j(\varPhi((I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k}))) + z^* \cdot \boldsymbol{k}_i. \end{aligned}$$

Analogously, we infer

$$\begin{aligned} \boldsymbol{C}_j(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k'}) &= \boldsymbol{s}_j(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k'}) - sk \cdot \boldsymbol{k}'_i \\ &= \boldsymbol{C}_j(\varPhi(I, (\omega_\mathsf{M}, \boldsymbol{r}), \boldsymbol{k'})) + z^* \cdot \boldsymbol{k}'_i. \end{aligned}$$

Combining these equations, we obtain:

$$\begin{aligned}&\boldsymbol{C}_j(\varPhi(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k}))+z^*\cdot\boldsymbol{k}_i\\&=\boldsymbol{C}_j(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k})=\boldsymbol{C}_j(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k'})\\&=\boldsymbol{C}_j(\varPhi(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k'}))+z^*\cdot\boldsymbol{k}'_i.\end{aligned}$$

Since 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

$$\begin{aligned}&\boldsymbol{C}_j(\varPhi(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k}))\\&=\boldsymbol{C}_j(\omega_{\mathsf{A}},\boldsymbol{r}-z^*\boldsymbol{c}(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k}),(sk+z^*,par),\boldsymbol{k})\\&=\boldsymbol{C}_j(\omega_{\mathsf{A}},\boldsymbol{r}-z^*\boldsymbol{d},I,\boldsymbol{k})\\&=\boldsymbol{C}_j(\omega_{\mathsf{A}},\boldsymbol{r}-z^*\boldsymbol{d},I,\boldsymbol{k'})\\&=\boldsymbol{C}_j(\omega_{\mathsf{A}},\boldsymbol{r}-z^*\boldsymbol{c}(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k'}),\boldsymbol{g},(sk+z^*,par),\boldsymbol{k'})\\&=\boldsymbol{C}_j(\varPhi(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k'})),\end{aligned}$$

where we have again used that $\hat{\boldsymbol{J}}j(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k})=\hat{\boldsymbol{J}}j(\varPhi(I,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{k}))=i$ and $\boldsymbol{k}|_i=\boldsymbol{k}'|_i$ together imply that $\boldsymbol{C}_j(I,(\omega_{\mathsf{M}},\boldsymbol{r}-z^*\boldsymbol{d}),\boldsymbol{k})=\boldsymbol{C}_j(I,(\omega_{\mathsf{M}},\boldsymbol{r}-z^*\boldsymbol{d}),\boldsymbol{k'})$. By combining equations, it now follows that $z^*\cdot\boldsymbol{k}_i=z^*\cdot\boldsymbol{k}'_i$ or, equivalently, $z^*\cdot(\boldsymbol{k}_i-\boldsymbol{k}'_i)=0$. Thus, torsion-freeness of z^* implies that $\boldsymbol{k}_i=\boldsymbol{k}'_i$ which contradicts the assumption that $\boldsymbol{k}_i\neq\boldsymbol{k}'_i$. This completes the proof.

The following lemma lower bounds the probability of $\mathcal{W}_{inj}$. Let $\mathcal{D}_{sk}$ and $\mathcal{D}_r$ denote the sets of secret keys and r's, respectively; then $\mathcal{E}=\mathcal{D}_{sk}\times\mathcal{D}_r^{Q_{\mathsf{P}_1}\eta}$. Let $\mathcal{E}_{inj}\subset\mathcal{E}$ s.t. for all $c\in\mathcal{S}_c$, $(sk,\boldsymbol{r})\in\mathcal{E}_{inj}\implies(sk+z^*,\boldsymbol{r}-c\cdot z^*)\in\mathcal{E}$. Accordingly, $\forall(sk,\boldsymbol{r})\in\mathcal{E}\setminus\mathcal{E}_{inj}:(sk+z^*,\boldsymbol{r}-c\cdot z^*)\notin\mathcal{E}$. Since $\varPhi$ maps sk to $sk+z^*$ and maps $\boldsymbol{r}$ to $\boldsymbol{r}-c(I,\omega,\boldsymbol{h})\cdot z^*$, $(sk,\boldsymbol{r})\in\mathcal{E}_{inj}\wedge(sk,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{h})\in\mathcal{W}$ implies that $(sk,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{h})\in\mathcal{W}_{inj}$.

Lemma 14. $\Pr_{(sk,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{h})\xleftarrow{\$}(\mathcal{I}\times\Omega\times\mathcal{C}^{Q_{\mathsf{V}}})}[(sk,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{h})\in\mathcal{W}_{inj}]\geq\varepsilon^2/8.$

Proof. In Lemma 9, we set $\mathcal{X}_1:=\mathcal{D}_{sk}$ and $\mathcal{X}_3:=\mathcal{D}_r^{Q_{\mathsf{P}_1}\eta}$ and accordingly, $\mathcal{S}:=\{1,3\}$. Then by Lemma 9, there exists a set $\mathcal{B}_{\mathcal{S},\varepsilon/2}$ s.t.

$$\Pr_{(sk,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{h})\xleftarrow{\$}(\mathcal{I}\times\Omega\times\mathcal{C}^{Q_{\mathsf{V}}})}[(sk,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{h})\in\mathcal{B}_{\mathcal{S},\varepsilon/2}]\geq\varepsilon/2$$

and s.t. for all $(sk,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{h})\in\mathcal{B}_{\mathcal{S},\varepsilon/2}$,

$$\Pr_{(sk',\boldsymbol{r}')\xleftarrow{\$}\mathcal{E}}\left[\big(sk',(\omega_{\mathsf{M}},\boldsymbol{r}'),\boldsymbol{h}\big)\in\mathcal{W}\right]\geq\varepsilon/2.$$

The latter inequality can be rewritten as

$$\begin{aligned}&\Pr_{(sk',\boldsymbol{r}')\xleftarrow{\$}\mathcal{E}}[\big(sk',(\omega_{\mathsf{M}},\boldsymbol{r}'),\boldsymbol{h}\big)\in\mathcal{W}\wedge(sk',\boldsymbol{r}')\in\mathcal{E}\setminus\mathcal{E}_{inj}] &\quad(18)\\&+\Pr_{(sk',\boldsymbol{r}')\xleftarrow{\$}\mathcal{E}}[\big(sk',(\omega_{\mathsf{M}},\boldsymbol{r}'),\boldsymbol{h}\big)\in\mathcal{W}\wedge(sk',\boldsymbol{r}')\in\mathcal{E}_{inj}]\\&=\Pr_{(sk',\boldsymbol{r}')\xleftarrow{\$}\mathcal{E}}[\big(sk',(\omega_{\mathsf{M}},\boldsymbol{r}'),\boldsymbol{h}\big)\in\mathcal{W}]\geq\varepsilon/2. &\quad(19)\end{aligned}$$

By assumption, we have $(\varepsilon,Q_{\mathsf{P}_1}\eta')$-regularity of F, and so at most an $\varepsilon/4$- fraction of $(sk,\boldsymbol{r}')\in\mathcal{E}$ satisfy $(sk,\boldsymbol{r}')\in\mathcal{E}\setminus\mathcal{E}_{inj}$. Hence, it follows that, for all $(I,\omega,\boldsymbol{h})=(sk,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{h})\in\mathcal{B}_{\mathcal{S},\varepsilon/2}$,

$$\begin{aligned}&\Pr_{(sk',\boldsymbol{r}')\xleftarrow{\$}\mathcal{E}}[\big(sk',(\omega_{\mathsf{M}},\boldsymbol{r}'),\boldsymbol{h}\big)\in\mathcal{W}\wedge(sk',\boldsymbol{r}')\in\mathcal{E}\setminus\mathcal{E}_{inj}]\\&\leq\Pr_{(sk',\boldsymbol{r}')\xleftarrow{\$}\mathcal{E}}[(sk',\boldsymbol{r}')\in\mathcal{E}\setminus\mathcal{E}_{inj}]\leq\varepsilon/4.\end{aligned}$$

By inequality 19, we obtain that for all $(I, \omega, \boldsymbol{h}) = (sk, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) \in \mathcal{B}_{\mathcal{S},\varepsilon/2}$,

$$\begin{aligned}
&\Pr_{(sk',\boldsymbol{r}')\xleftarrow{\$}\mathcal{E}}[\big(sk', (\omega_{\mathsf{M}}, \boldsymbol{r}'), \boldsymbol{h}\big) \in \mathcal{W}_{inj}] + \varepsilon/4 \\
\geq &\Pr_{(sk',\boldsymbol{r}')\xleftarrow{\$}\mathcal{E}}[\big(sk', (\omega_{\mathsf{M}}, \boldsymbol{r}'), \boldsymbol{h}\big) \in \mathcal{W} \wedge (sk', \boldsymbol{r}') \in \mathcal{E}_{inj}] + \varepsilon/4 \\
\geq &\Pr_{(sk',\boldsymbol{r}')\xleftarrow{\$}\mathcal{E}}[\big(sk', (\omega_{\mathsf{M}}, \boldsymbol{r}'), \boldsymbol{h}\big) \in \mathcal{W} \wedge (sk', \boldsymbol{r}') \in \mathcal{E}_{inj}] \\
+ &\Pr_{(sk',\boldsymbol{r}')\xleftarrow{\$}\mathcal{E}}[\big(sk', (\omega_{\mathsf{M}}, \boldsymbol{r}'), \boldsymbol{h}\big) \in \mathcal{W} \wedge (sk', \boldsymbol{r}') \in \mathcal{E} \setminus \mathcal{E}_{inj}] \\
= &\Pr_{(sk',\boldsymbol{r}')\xleftarrow{\$}\mathcal{E}}[\big(sk', (\omega_{\mathsf{M}}, \boldsymbol{r}'), \boldsymbol{h}\big) \in \mathcal{W}] \geq \varepsilon/2,
\end{aligned}$$

which implies that for all $(I, \omega, \boldsymbol{h}) = (sk, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) \in \mathcal{B}_{\mathcal{S},\varepsilon/2}$,

$$\Pr_{(sk',\boldsymbol{r}')\xleftarrow{\$}\mathcal{E}}[\big(sk', (\omega_{\mathsf{M}}, \boldsymbol{r}'), \boldsymbol{h}\big) \in \mathcal{W}_{inj}] \geq \varepsilon/4.$$

Putting things together, we see that

$$\begin{aligned}
&\Pr_{(sk,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{h})\xleftarrow{\$}(\mathcal{I}\times\Omega\times\mathcal{C}^{Q_{\mathsf{V}}})}[(sk, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) \in \mathcal{W}_{inj}] \\
= &\Pr_{(sk,(\omega_{\mathsf{M}},\boldsymbol{r}),\boldsymbol{h})\xleftarrow{\$}(\mathcal{I}\times\Omega\times\mathcal{C}^{Q_{\mathsf{V}}}),(sk',\boldsymbol{r}')\xleftarrow{\$}\mathcal{E}}[\big(sk', (\omega_{\mathsf{M}}, \boldsymbol{r}'), \boldsymbol{h}\big) \in \mathcal{W}_{inj}] \\
= &\sum_{(\widehat{I},\widehat{\omega},\widehat{\boldsymbol{h}})\in(\mathcal{I}\times\Omega\times\mathcal{C}^{Q_{\mathsf{V}}})} \Pr\left[\begin{array}{l}\big(sk', (\omega_{\mathsf{M}}, \boldsymbol{r}'), \boldsymbol{h}\big) \in \mathcal{W}_{inj} \wedge \\ (sk, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) = (\widehat{I}, \widehat{\omega}, \widehat{\boldsymbol{h}})\end{array}\right] \\
\geq &\sum_{(\widehat{I},\widehat{\omega},\widehat{\boldsymbol{h}})\in\mathcal{B}_{\mathcal{S},\varepsilon/2}} \Pr\left[\begin{array}{l}\big(sk', (\omega_{\mathsf{M}}, \boldsymbol{r}'), \boldsymbol{h}\big) \in \mathcal{W}_{inj} \wedge \\ (sk, (\omega_{\mathsf{M}}, \boldsymbol{r}), \boldsymbol{h}) = (\widehat{I}, \widehat{\omega}, \widehat{\boldsymbol{h}})\end{array}\right] \\
= &\sum_{(\widehat{I},\widehat{\omega},\widehat{\boldsymbol{h}})\in\mathcal{B}_{\mathcal{S},\varepsilon/2}} \Pr_{(sk',\boldsymbol{r}')\xleftarrow{\$}\mathcal{E}}\left[\Big(sk', (\widehat{\omega_{\mathsf{M}}}, \boldsymbol{r}'), \widehat{\boldsymbol{h}}\Big) \in \mathcal{W}_{inj}\right] \\
\cdot &\Pr_{(I,\omega,\boldsymbol{h})\xleftarrow{\$}(\mathcal{I}\times\Omega\times\mathcal{C}^{Q_{\mathsf{V}}})}[(I, \omega, \boldsymbol{h}) = (\widehat{I}, \widehat{\omega}, \widehat{\boldsymbol{h}})] \\
\geq &\varepsilon/4 \cdot \sum_{(\widehat{I},\widehat{\omega},\widehat{\boldsymbol{h}})\in\mathcal{B}_{\mathcal{S},\varepsilon/2}} \Pr_{(I,\omega,\boldsymbol{h})\xleftarrow{\$}(\mathcal{I}\times\Omega\times\mathcal{C}^{Q_{\mathsf{V}}})}[(I, \omega, \boldsymbol{h}) = (\widehat{I}, \widehat{\omega}, \widehat{\boldsymbol{h}})] \\
= &\varepsilon/4 \cdot \Pr_{(I,\omega,\boldsymbol{h})\xleftarrow{\$}(\mathcal{I}\times\Omega\times\mathcal{C}^{Q_{\mathsf{V}}})}[(I, \omega, \boldsymbol{h}) \in \mathcal{B}_{\mathcal{S},\varepsilon/2}] \geq \varepsilon/4 \cdot \varepsilon/2 = \varepsilon^2/8.
\end{aligned}$$

We show in Lemma 7 that it is possible to construct sets $\mathcal{E}$ and $\mathcal{E}_{inj}$ which satisfy the assumption in Lemma 14.

Lemma 15. $\Pr_{(I,\omega,\boldsymbol{h})\xleftarrow{\$}(\mathcal{I}\times\Omega\times\mathcal{C}^{Q_{\mathsf{V}}})}[(I, \omega, \boldsymbol{h}) \in \mathcal{B}] \geq \frac{1}{2}\left(\varepsilon^2/8 - \frac{Q_{\mathsf{V}}^{Q_{\mathsf{P}_2}+1}\cdot\binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{\left(\frac{|\mathcal{C}|}{|\mathcal{S}_c|}\right)^{Q_{\mathsf{P}_2}}\cdot|\mathcal{C}|}\right).$

Proof. In the following, let $\mathcal{G}_{inj} = \mathcal{W}_{inj} \cap \mathcal{G}$ and $\mathcal{B}_{inj} = \mathcal{W}_{inj} \cap \mathcal{B}$. Since for all $(I, \omega, \boldsymbol{h}) \in \mathcal{W}_{inj}$, we have $\Phi(I, \omega, \boldsymbol{h}) = \Phi_{inj}(I, \omega, \boldsymbol{h}) \in \mathcal{W} = \mathcal{G} \cup \mathcal{B}$, we can partition $\mathcal{G}_{inj}$ into subsets $\mathcal{G}_{inj}^g$ and $\mathcal{G}_{inj}^b$, such that all elements in $\mathcal{G}_{inj}^g$ are mapped into $\mathcal{G}$ via Φ_{inj} and all elements in $\mathcal{G}_{inj}^b$ are mapped into $\mathcal{B}$ via Φ_{inj}. It follows that

$$\begin{aligned}
&\Pr_{(I,\omega,\boldsymbol{h})\xleftarrow{\$}(\mathcal{I}\times\Omega\times\mathcal{C}^{Q_{\mathsf{V}}})}[(I, \omega, \boldsymbol{h}) \in \mathcal{G}_{inj}] \\
= &\Pr_{(I,\omega,\boldsymbol{h})\xleftarrow{\$}(\mathcal{I}\times\Omega\times\mathcal{C}^{Q_{\mathsf{V}}})}[(I, \omega, \boldsymbol{h}) \in \mathcal{G}_{inj}^g] + \Pr_{(I,\omega,\boldsymbol{h})\xleftarrow{\$}(\mathcal{I}\times\Omega\times\mathcal{C}^{Q_{\mathsf{V}}})}[(I, \omega, \boldsymbol{h}) \in \mathcal{G}_{inj}^b] \qquad (20)
\end{aligned}$$

By Lemma **??**, we have:

$$\Pr_{(I,\omega,\boldsymbol{h})\xleftarrow{\$}(\mathcal{I}\times\Omega\times\mathcal{C}^{Q_{\mathsf{V}}})}[(I, \omega, \boldsymbol{h}) \in \mathcal{G}_{inj}^g] \leq \frac{Q_{\mathsf{V}}^{Q_{\mathsf{P}_2}+1}\cdot\binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{\left(\frac{|\mathcal{C}|}{|\mathcal{S}_c|}\right)^{Q_{\mathsf{P}_2}}\cdot|\mathcal{C}|}. \qquad (21)$$

Because Φ_{inj} is injective:

$$\Pr_{(I,\omega,\boldsymbol{h})\xleftarrow{\$}(\mathcal{I}\times\Omega\times\mathcal{C}^{Q_{\mathsf{V}}})}[(I,\omega,\boldsymbol{h})\in\mathcal{G}^{b}_{inj}] \leq \Pr_{(I,\omega,\boldsymbol{h})\xleftarrow{\$}(\mathcal{I}\times\Omega\times\mathcal{C}^{Q_{\mathsf{V}}})}[(I,\omega,\boldsymbol{h})\in\mathcal{B}]. \tag{22}$$

It follows from 20,21, and 22 that

$$\Pr[(I,\omega,\boldsymbol{h})\in\mathcal{G}_{inj}] \leq \frac{Q_{\mathsf{V}}^{Q_{\mathsf{P}_2}+1}\cdot\binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{\left(\frac{|\mathcal{C}|}{|\mathcal{S}_c|}\right)^{Q_{\mathsf{P}_2}}\cdot|\mathcal{C}|} + \Pr[(I,\omega,\boldsymbol{h})\in\mathcal{B}].$$

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

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

Since by the previous Lemma, $\Pr[(I,\omega,\boldsymbol{h})\in\mathcal{W}_{inj}] = \varepsilon^2/8$, we finally obtain

$$\Pr_{(I,\omega,\boldsymbol{h})\xleftarrow{\$}(\mathcal{I}\times\Omega\times\mathcal{C}^{Q_{\mathsf{V}}})}[(I,\omega,\boldsymbol{h})\in\mathcal{B}] \geq \frac{1}{2}\left(\varepsilon^2/8 - \frac{Q_{\mathsf{V}}^{Q_{\mathsf{P}_2}+1}\cdot\binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{\left(\frac{|\mathcal{C}|}{|\mathcal{S}_c|}\right)^{Q_{\mathsf{P}_2}}\cdot|\mathcal{C}|}\right).$$

We are now ready to prove Lemma 11, i.e., we show that there exist $i\in[Q_{\mathsf{P}_2}+1], j\in[Q_{\mathsf{V}}]$ such that $\beta_{i,j} > \left(\frac{\varepsilon^2}{8} - \frac{Q_{\mathsf{V}}^{Q_{\mathsf{P}_2}+1}\cdot\binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{\left(\frac{|\mathcal{C}|}{|\mathcal{S}_c|}\right)^{Q_{\mathsf{P}_2}}\cdot|\mathcal{C}|}\right)\cdot\frac{1}{2Q_{\mathsf{V}}(Q_{\mathsf{P}_2}+1)}$. Toward a contradiction, suppose instead that for all $i\in[Q_{\mathsf{P}_2}+1], j\in[Q_{\mathsf{V}}]$, we have that

$$\Pr_{(I,\omega,\boldsymbol{h})\xleftarrow{\$}(\mathcal{I}\times\Omega\times\mathcal{C}^{Q_{\mathsf{V}}})}[(I,\omega,\boldsymbol{h})\in\mathcal{B}_{i,j}] < \left(\frac{\varepsilon^2}{8} - \frac{Q_{\mathsf{V}}^{Q_{\mathsf{P}_2}+1}\cdot\binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{\left(\frac{|\mathcal{C}|}{|\mathcal{S}_c|}\right)^{Q_{\mathsf{P}_2}}\cdot|\mathcal{C}|}\right)\cdot\frac{1}{2Q_{\mathsf{V}}(Q_{\mathsf{P}_2}+1)}.$$

By Lemma 15,

$$\begin{aligned}&\frac{1}{2}\left(\frac{\varepsilon^2}{8} - \frac{Q_{\mathsf{V}}^{Q_{\mathsf{P}_2}+1}\cdot\binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{\left(\frac{|\mathcal{C}|}{|\mathcal{S}_c|}\right)^{Q_{\mathsf{P}_2}}\cdot|\mathcal{C}|}\right) \leq \Pr[(I,\omega,\boldsymbol{h})\in\mathcal{B}] = \Pr[(I,\omega,\boldsymbol{h})\in\bigcup_{i,j}\mathcal{B}_{i,j}]\\ &\leq \sum_{i,j}\Pr[(I,\omega,\boldsymbol{h})\in\mathcal{B}_{i,j}] < \frac{1}{2}\left(\frac{\varepsilon^2}{8} - \frac{Q_{\mathsf{V}}^{Q_{\mathsf{P}_2}+1}\cdot\binom{Q_{\mathsf{P}_2}+Q_{\mathsf{P}_1}}{Q_{\mathsf{P}_1}}}{\left(\frac{|\mathcal{C}|}{|\mathcal{S}_c|}\right)^{Q_{\mathsf{P}_2}}\cdot|\mathcal{C}|}\right).\end{aligned}$$

This is a contradiction.