Signatures with Tight Multi-User Security from Search Assumptions

Jiaxin Pan and Magnus Ringerud

Department of Mathematical Sciences
NTNU – Norwegian University of Science and Technology, Trondheim, Norway
{jiaxin.pan, magnus.ringerud}@ntnu.no

Abstract. We construct two tightly secure signature schemes based on the computational Diffie-Hellman (CDH) and factoring assumptions in the random oracle model. Our schemes are proven secure in the multi-user setting, and their security loss is constant and does not depend on the number of users or signing queries. They are the first schemes that achieve this based on standard search assumptions, as all existing schemes we are aware of are either based on stronger decisional assumptions, or proven tightly secure in the less realistic single-user setting. Under a concrete estimation, in a truly large scale, the cost of our CDH-based scheme is about half of Schnorr and DSA (in terms of signature size and running time for signing).

Keywords. Digital signature, tight reduction, multi-user security, search assumption.

1 Introduction

In modern public-key cryptography, a scheme is usually proposed together with a reduction-based security analysis. In such an analysis, a security model is defined to capture the security required in the real world. Then a reduction is constructed to show that if there is an adversary can break the security of the scheme, then the reduction can use this adversary to break some well-studied hardness assumption.

This analysis provides not only a mathematically proof for the security of a scheme, but also guidelines for theoretically sound parameter setup, namely, setting up parameters for a scheme so that it can offer the proven security guarantee.

Concrete Security. To deploy a scheme in a theoretically sound manner, we need to know the scheme's concrete security. The reduction-based analysis offers a way to do so. More precisely, it establishes the following relation between the success ratio $\Gamma_{\mathcal{A}}$ of an adversary $\mathcal{A}$ (which is defined as the quotient of its success probability and running time) attacking scheme S, and that of a reduction $\mathcal{B}$ breaking the underlying assumption P:

$$\Gamma_{\mathcal{A}} \leq L \cdot \Gamma_{\mathcal{B}}. \tag{1}$$

The parameter L is called the security loss. Equation (1) guides us in deriving parameters that can provably guarantee a *k-bit security* for the scheme[1]. According to the current cryptanalysis results, we derive suitable parameters for the hardness problem P to compensate the security loss L and have $\Gamma_{\mathcal{A}} \leq L \cdot \Gamma_{\mathcal{B}} \leq 2^{-k}$. Thus a smaller L can give us shorter key lengths, and potentially more efficient schemes.

We call a reduction (or the scheme's security) *tight* if L is a small constant. Recently, a relaxed notion, called almost tight security, was considered in [18,25,26], where L could be a linear or logarithmic function of the security parameter. In this paper, we only consider fully tight security. In non-tight schemes, the security loss can depend on the scale of applications, for instance the number of users and/or issued signatures for digital signatures. To provide the same level of security guarantee, one needs to reasonably estimate the scale of an application and derive larger parameters to compensate for the security loss. Such an increase in parameters will inevitably slow down computations.

Thus, a large amount of attention has recently been drawn towards research on tight security, which has spanned from theoretical (such as [31,13,4]) to more practical aspects (such as [27,20], and covered different primitives including (identity-based) encryption [25,18,14], digital signatures [28,31,30,26,27] and non-interactive zero-knowledge proofs [3,2].

In this paper we focus on digital signatures, which has numerous applications both on its own, and as a basic building block for advanced cryptographic protocols (for instance, TLS).

Multi-User Security. The classical security model (or definition) for signature schemes is unforgeability against chosen-message attacks (UF-CMA) [29], where an adversary attempts to forge a signature on a fresh message after it adaptively asks for signatures on multiple different messages. The UF-CMA security is defined in the *single-user* setting, namely, an adversary can only see the public key of a single user. We believe this is less desirable in practice.

In practice, (independent) public keys of multiple users are exposed to an adversary. Presumably, it will output a valid forgery under one of these public keys in a meaningful way after asking multiple signatures. This is captured by the UF-CMA security in the multi-user setting (denoted by MU-UF-CMA).

Although the MU-UF-CMA security is more desirable than the UF-CMA security, most signature schemes are typically proven in the UF-CMA model. We believe there are two main reasons: Firstly, adversaries in the UF-CMA model have less capabilities and thus the security proof in this model is easier; secondly, asymptotically speaking, the UF-CMA security implies MU-UF-CMA according to a generic reduction in [24]. However, this is problematic when we consider concrete security and derive theoretically sound parameters for the scheme in practice, since the generic reduction in [24] is not tight.

Concretely, it loses a factor of ℓ, which is the number of users: It only proves that attacking a scheme in the MU-UF-CMA model with ℓ users does not increase

[1] Usually, "k-bit security" means that there is no adversary can break the scheme with success ratio larger than 2^{-k} (see discussions in [7,17]).

the success ratio of the adversary by more than a factor of ℓ, compared to attacking the same scheme in the UF-CMA model. Thus, via this non-tight generic reduction, a signature scheme with k-bit security guarantee in the UF-CMA model does not gives us the same level of provable security guarantee in the MU-UF-CMA model.

As a concrete example, we reasonably assume $\ell := 2^{30}$ (about 1 billion)[2]. For a signature scheme, if the best adversary attacking it in the UF-CMA model has success ratio $\Gamma_{\mathsf{U}} := 2^{-80}$ (i.e. 80-bit UF-CMA security), then the argument in [24] shows that the best adversary against the same scheme in the MU-UF-CMA model has success ratio $\Gamma_{\mathsf{MU}} = 2^{30} \cdot 2^{-80} = 2^{-50}$, which is not a safe margin for current large-scale applications. To provide the same level of security in the MU-UF-CMA model, we need to increase the key length accordingly to compensate the security loss, which is $\ell := 2^{30}$ in the above case.

Difficulty: Tight Security from Search Assumptions. In recent years, several signature schemes with tight security in the single-user setting (aka. UF-CMA security) have been created, such as [28,31,30,18,14,13,34,26]. The schemes in [4,38,27,44,43] are the only ones we know of that have tight security in the multi-user setting (aka. MU-UF-CMA security). We note that [43] is based on the one-more CDH assumption, which is a non-static interactive assumption in pairing groups.

Furthermore, most of all the known tightly secure schemes (in both single-user and multi-user settings) require decisional assumptions. Inherently, decisional assumptions seem crucial for tight security. Different to the non-tight and guessing proof strategy, decisional assumptions and their random self-reducibility give security reductions the advantage to switch the distribution of signatures to random "at once", and then argue that even for an unbounded adversary there is no chance to win. This advantage cannot be easily achieved by search assumptions (such as the Computational Diffie-Hellman (CDH) and Factoring (FAC) assumptions), although search assumptions are more standard and reliable. For instance, the CDH assumption is more standard and weaker than the Decisional Diffie-Hellman assumption. It is similar for the FAC and the decisional Phi-Hiding assumption used in [34].

There are a few notable exceptions including the Rabin-William scheme [11] and the Micali-Reyzin scheme [40,6] based on FAC, the "selector bit" variants of RSA-PSS [35], and the Chevallier-Mames [19] and its later abstraction by Kiltz, Loss, and Pan [37]. However, their tight security is established in the less realistic single-user setting.

As a result of the above discussion, we raise the question of whether it is possible to construct an efficient and tightly MU-UF-CMA-secure signature scheme based on standard search assumptions. We are interested in schemes in the random oracle model [8]. In the random oracle model, a cryptographic hash function is modeled as an oracle that responds a random value in its output domain for each unique query. Although there is some limitation with the

[2] Nowadays many applications involve billions of users. For instance, Facebook has about 2 billion active users daily, according to https://about.fb.com/company-info/.

model [16], security proofs in the model still give strong evidence of the scheme's practical security. Moreover, schemes in the random oracle model are usually more efficient than their counterparts in the standard model.

1.1 Our Contribution: Multi-User Security from Search Assumptions

We construct two tightly secure signature schemes from standard (static) search assumptions (namely, CDH and FAC) in the multi-user setting. The security is proven in the random oracle model and the security loss is the constant 1. Our schemes improve upon those from the framework of Kiltz, Loss, and Pan at Asiacrypt 2017 [37] in the sense that our schemes have tight multi-user security. Asymptotically, our schemes have the same number of elements in a signature as [37], but, since our schemes are tightly secure in the multi-user setting, at the concrete security level our elements will be shorter and our schemes will have smaller signature size and achieve more efficient computation, in particular, for settings with large number of users. In fact, our CDH-based scheme is the Chevallier-Mames scheme [19]. Another interpretation of it is that we give a new tight security proof of the original Chevallier-Mames scheme in the multi-user setting.

In the following efficiency analysis, it shows that our CDH-based scheme is more efficient than Schnorr and DSA in a truly large setting. Moreover, our CDH-based scheme can offer offline pre-computation to speed up signing, namely, most of the work can be done offline before receiving the signing messages.

Efficiency Analysis. We compare the asymptotic efficiency of known tightly secure signature schemes (in both single-user and multi-user settings) in the random oracle model in Table 1. We are precise about the security loss from the single-user to the multi-user setting. The multi-user security of some schemes is established by the non-tight reduction in [24] and thus we need to choose a larger group to compensate the non-trivial security loss. We will mark those group sizes with G_ℓ. We also include the two famous signature schemes Schnorr and DSA in our comparison. By the optimal security proof in [38], the security loss of Schnorr is $12Q_h$, where Q_h is the number of hash queries an adversary makes, and the loss of the Katz-Wang scheme (KW) [28] is 4. We note a recent work on Schnorr in the (idealized) generic group model (GGM) [15]. While a proof in the GGM certainly provides certain degree of confidence in the scheme's security, its scope is rather limited, for instance, it does not capture algorithms that make use of the representation of the group. Thus, we do not include their result in our comparison. The provable security result for DSA [36] is established by [22] in the single-user setting, and we believe it is hard to prove it tightly in the multi-user setting. We will give more details about this in Appendix A.

To provide the concrete efficiency comparison, we estimate the schemes based on the DLOG and Diffie-Hellman assumptions in Table 2. We consider exponentiation as the dominating factor in the running time cost. We use elliptic curves when estimating the schemes, as group elements have a much shorter

Scheme	Approx. Size	Off-line Exp.	On-line Exp.	Loss	Ass.	Search?
Schnorr [42]	$n + \|p\|$	1	0	$12Q_h$	DLOG	✓
DSA [36]	$2\|p\|$	1	0	ℓQ_h	DLOG	✓
KW [28,38]	$n + \|p\|$	2	0	4	DDH	✗
GJKW [28]	$G_\ell + n + \|p\|$	1	2	ℓ	CDH	✓
$\mathsf{FS}_{\mathsf{CDH}}$ [37]	$G_\ell + n + \|p\|$	1	2	ℓ	CDH	✓
$\mathsf{OF}_{\mathsf{CDH}}$ [37]	$G_\ell + n + \|p\|$	3	0	ℓ	CDH	✓
AFLT [1]	$2n + c$	1	0	ℓ	DSDL	✗
$\mathsf{FS}_{\mathsf{SCDH}}$ [37]	$G_\ell + 2n + c$	1	2	ℓ	SCDH	✓
$\mathsf{OF}_{\mathsf{SCDH}}$ [37]	$G_\ell + 2n + c$	3	0	ℓ	SCDH	✓
GJ [27]	$2G + n + 4\|p\|$	0	7	3	CDH&DDH	✗
WLGSZ [43]	$2G' + 1$	1	1	1	OMCDH	(✓)
Ours (Fig. 2)	$G + n + \|p\|$	3	0	1	CDH	✓
MR [40, §4.3]	$n + \|N\|$	1	1	ℓ	FAC	✓
BR [9]	$n + \|N\|$	0	0	ℓ	FAC	✓
RSA-FDH [33]	$\|N\|$	0	1	ℓ	ΦH	✗
$\mathsf{FS}_{\mathsf{FAC}}$ [37]	$G_\ell + n + \|N\|$	1	2	ℓ	FAC	✓
Ours (Fig. 4)	$G + n + \|N\|$	1	2	1	FAC	✓

Table 1. Comparison between some known signature schemes in the random oracle model. Top: schemes in a cyclic group $\mathbb{G}$ of prime order p. Bottom: schemes over $\mathbb{Z}_N$ for composite N. We detail the security loss of the schemes in the multi-user setting with ℓ users. Q_h is the maximum number of hash queries an adversary can make. Elements of $\mathbb{G}$ have bit length G and n denotes the security parameter. We take the security loss into account, and, for non-tight schemes, we write their group size as G_ℓ. G' denotes the bit length of a pairing-friendly group. $c < |p|$ is a parameter for the short Diffie-Hellman assumptions. We count the numbers of offline ("Off-line Exp.") and online exponentiation ("On-line Exp.") during signing, respectively.

representation there than over finite fields. To have a k-bit secure DLOG problem, we need to choose a $2k$-bit elliptic curve, according to the baby-step giant-step algorithm. As in [27], we assume $k + 1$ bits to represent a k-bit elliptic curve group element, and k bits to represent the corresponding discrete log. Thus, we need 257 bits to represent a group element of the NIST P256 curve.

For the running time in Table 2, similar to [27], we run "`openssl speed ecdh`" on a computer with a 2.4 GHz Quad-Core Intel Core i5 CPU, 16 GB RAM and MacOS 10.15.3. This command offers speed estimation for one operation (namely, exponentiation in the language of this paper) for curves NIST P192 (takes 0.3 milliseconds), P224 (0.4ms), P256 (0.4ms), P384 (1.0ms), P521 (2.2ms), K233 (2.6ms), B163 (1.3ms) and so on. We use NIST P-curves for estimation, as they are more efficient than the other curves providing the same security level. We note that the security of Schnorr and DSA is dependent on Q_h, which is problematic, since an adversary can compute as many hash values as he would like offline. Computing hash functions is very cheap, and for instance, one can easily compute 2^{29} (≈0.5 billion) SHA-512 of 8192 byte messages per second with a normal PC. This is estimated by running "`openssl speed sha`". Thus, Q_h can be much larger than the number of users. According to [38], Q_h is estimated

in the range between 2^{40} to 2^{80}. We consider a setting with roughly a billion users ($\ell := 2^{30}$), and take DSA as an example to show how we estimate: For $(\ell, Q_h) = (2^{30}, 2^{40})$ and 128-bit security, the security loss is 2^{70}, and we require a 198-bit secure DLOG. Thus we need a 396-bit curve, and we suggest NIST P521 as the appropriate choice, for which one signing (which requires 1 operation) takes 2.2ms.

We note that the WLGSZ scheme uses Type 1 (symmetric) pairings [23]. Usually, in pairing-friendly groups, the group size is larger and operations (in particular, computing pairings) are less efficient than those in groups without pairings. For 128-bit security of WLGSZ, we should choose a Supersingular Curve over $GF(2^{1223})$, where 1 group operation takes 2.57ms, and 1 pairing takes 19.00ms.[3] We also put the estimation of it in Table 2.

Scheme	Q_h	Curve	Sig. Size (in bits)	Sig. Time (in milliseconds)
Schnorr [42]	2^{40}	P384	768	1.0
Schnorr [42]	2^{80}	P521	1024	2.2
DSA [36]	2^{40}	P521	1024	2.2
DSA [36]	2^{80}	P521	1024	2.2
GJKW [28]	–	P384	1153	3.0
$\mathsf{FS}_{\mathsf{CDH}}$ [37]	–	P384	1153	3.0
$\mathsf{OF}_{\mathsf{CDH}}$ [37]	–	P384	1153	3.0
WLGSZ [43]	–	SS	2447	5.14
Ours (Fig. 2)	–	P256	769	1.2
KW [28,38]	–	P256	512	0.4
GJ [27]	–	P256	1794	2.8

Table 2. Concrete efficiency estimation of some known signature schemes based on the DLOG-related assumptions for 128-bit security and 2^{30} (≈ 1 billion) users. Top: schemes using search assumptions. Bottom: schemes using decisional assumptions. For the same security level, we focus on the signature size ("Sig. Size") and running time for signing ("Sig. Time"). '–' means the security of the corresponding scheme is independent of that parameter.

Interpretation and Open Problems. According to Table 2, for a medium scale ($(\ell, Q_h) = (2^{30}, 2^{40})$), our scheme based on CDH (cf. $\mathsf{PF\text{-}OF}_{\mathsf{CDH}}$ in Figure 2) is comparable to the Schnorr signature, but for a truly large scale ($(\ell, Q_h) = (2^{30}, 2^{80})$) our scheme is significantly more efficient than other schemes based on search assumptions (either DLOG or CDH).

It is worth mentioning that the KW scheme achieves the best efficiency at the cost of using a stronger assumption (DDH). CDH is more standard and weaker than DDH. For instance, in symmetric pairing groups, CDH is still hard, while DDH is easy. In fact, for certain primes, CDH is equivalent to DLOG [21,39].

[3] Taken from the benchmarks in https://github.com/miracl/MIRACL/blob/master/docs/miracl-explained/benchmarks.md (2020-03-26)

Our schemes live in harmony with the existing impossibility results about tightness [5,41,20]. Firstly, our schemes are not unique with respect to [5, Definition 1] and [41, Definition 1], and thus we do not contradict their results. Secondly, Cohn-Gordon et al.[20] showed the tightness impossibility result about authenticated key exchange protocols in a model where an adversary is allowed to corrupt a user's secret key, while our model does not allow signing key corruptions. This is a disadvantage of our schemes, since if one combines our schemes with the framework in [27] to construct an AKE protocol, the resulting protocol cannot provide any tight forward secrecy. We leave improving our schemes to allow signing key corruptions in a tight manner as the main open problem.

Another natural open problem is to further improve the efficiency of our schemes.

Our Approach. We provide a brief overview of our technique. The starting point of our work is the work of Kiltz, Loss, and Pan (KLP) [37], which tightly transforms a five-move identification (ID) scheme into a signature scheme with programmable random oracles in the single-user setting. Before them, a similar work of Kiltz, Masny, and Pan (KMP) [38] has been done for the three-move identification schemes in the *multi-user* setting, and the Schnorr signature is a well-known example from this transformation. In particular, the KMP framework proves that the UF-KOA security implies the MU-UF-CMA security for signatures (cf. Appendix B and Theorem 3.2 in [38]). The UF-KOA security is the same as UF-CMA, except that an adversary cannot ask any signing queries. Naturally, one is tempted to transform the single-user security (UF-KOA) to multi-user (MU-UF-CMA) one for KLP signatures by using the KMP method.

In the "UF-KOA $\rightarrow$ MU-UF-CMA" for SIG[ID] [4], the security reduction gets a public key pk from the UF-KOA challenger. By the random self-reducibility (RSR) of ID, the reduction can randomize pk and derive public keys $(pk_1, ..., pk_\ell)$, which is given to the adversary $\mathcal{A}$ against the MU-UF-CMA security. Due to some technical reason, only about half of $(pk_1, ..., pk_\ell)$ are computed using pk and the other half are generated honestly. Signing queries from $\mathcal{A}$ is generated by the honest-verifier zero-knowledge property of ID and programming the random oracle. To correctly map a MU-UF-CMA forgery to a UF-KOA one, the RSR property of ID allows, given the randomization trapdoor τ_i (for a $1 \leq i \leq \ell$), a valid transcript $\mathbf{t}_1 := (R, h, s)$ under pk_i to be turned into another valid transcript $\mathbf{t}_2 := (R, h, s^*)$ under $\boxed{pk}$. The reduction crucially requires that only the value s^* in $\mathbf{t}_2$ is different to s in $\mathbf{t}_1$.

The five-move ID schemes in KLP only have a weaker form of RSR, namely, given τ_i, a valid transcript (R_1, h_1, R_2, h_2, s) under pk_i can be converted to another valid transcript $(R_1, h_1, \boxed{R_2'}, h_2, s^*)$ under pk for $R_2 \neq R_2'$, since R_2 is dependent of pk_i. Unfortunately, this is problematic for converting a valid MU-UF-CMA forgery to a UF-KOA one: For a valid UF-KOA forgery under pk, h_2 has to be equal to $H_2(R_2', m)$ and, in particular, H_2 is simulated by the UF-KOA

[4] SIG[ID] is the signature scheme constructed from a three-move identification scheme ID via the Fiat-Shamir transformation.

challenger; However, before the reduction receives $\mathcal{A}$'s MU-UF-CMA forgery of message m under public key pk_i, h_2 has been defined as $h_2 := H_2(R_2, m)$ in one of the random oracle queries. Clearly, $H_2(R_2, m) \neq H_2(R_2', m)$ with overwhelming probability. Our solution is to apply the key-prefixing technique [12] and append R_1 and a public key in H_2, namely, we compute $H_2(R_1, R_2, pk, m)$ in our schemes. By knowing this additional information, we can carefully modify how the reduction queries the random oracle H_2, and make sure that $h_2 = H_2(R_1, R_2', pk, m)$. We will refer to Sections 3 and 4 for technical details.

2 Preliminaries

NOTATIONS. For a prime p, $\mathbb{Z}_p$ is the residual ring $\mathbb{Z}/p\mathbb{Z}$. If A is a set, then $a \xleftarrow{\$} A$ denotes picking a from A according to the uniform distribution. All our algorithms are probabilistic polynomial time, otherwise, we will state it. Let A be an algorithm and $a \xleftarrow{\$} \mathsf{A}(b)$ denote the output of A on input b.

We present our definitions and proofs in the code-based game-playing framework [10,14]. A game G contains procedures INITIALIZE and FINALIZE, and some additional procedures $\mathrm{P}_1, \ldots, \mathrm{P}_n$, which are defined in pseudo-code. Initially all variables in a game are undefined (denoted by $\perp$) and all sets are empty (denoted by $\emptyset$). An adversary $\mathcal{A}$ is executed in game G (denoted by $\mathsf{G}^{\mathcal{A}}$) if it first calls INITIALIZE, obtaining its output. Next, it may make arbitrary queries to P_i (according to their specification), again obtaining their output. Finally, it makes one single call to FINALIZE$(\cdot)$ and stops. We use $\mathsf{G}^{\mathcal{A}} \Rightarrow d$ to denote that G outputs d after interacting with $\mathcal{A}$, and d is the output of FINALIZE.

2.1 The Computational Diffie-Hellman Assumption

A cyclic group generator $\mathcal{G}$ is an algorithm that takes 1^n as input (where n is the security parameter), and returns a n-bit prime p, a cyclic group $\mathbb{G}$ of order p, and a generator of the group. We denote the output as $(p, g, \mathbb{G}) \xleftarrow{\$} \mathcal{G}(1^n)$.

Definition 1 (Computational Diffie-Hellman Assumption). *The computational Diffie-Hellman problem* CDH *is (t, ε)-hard with respect to $\mathcal{G}$ if for all adversaries $\mathcal{A}$ running in time at most t, we have*

$$\Pr[Z = g^{xy} \mid \mathsf{par} := (p, g, \mathbb{G}) \xleftarrow{\$} \mathcal{G}(1^n); x, y \xleftarrow{\$} \mathbb{Z}_p, Z \leftarrow \mathcal{A}(\mathsf{par}, g^x, g^y)] \leq \varepsilon.$$

2.2 The Factoring Assumption

The factoring-based scheme in [37] is proven based on the CDH assumption in the group of signed quadratic residues [32], which is tightly implied by the factoring assumption. We recall necessary background here. It is almost verbatim to the definitions in Section 4.3 of [37].

For $n \in \mathbb{N}$, we denote $\mathbb{P}_{n/2}$ as the set of $n/2$ bit primes, and $\mathsf{Blum}_n := \{N \mid N = (2p+1)(2q+1) \wedge (2p+1), (2q+1), p, q \in \mathbb{P}_{n/2} \wedge p \neq q\}$. The factoring assumption is defined as follows.

Definition 2 (Factoring Assumption). *The factoring problem* FAC *is* (t, ε) *hard for* Blum_n *if for all adversaries* $\mathcal{A}$ *running in time at most* t*,*

$$\Pr\left[N = PQ \wedge P, Q \in \mathbb{P}_{n/2} \mid N \xleftarrow{\$} \mathsf{Blum}_n; (P, Q) \leftarrow \mathcal{A}(N)\right] \leq \varepsilon. \tag{2}$$

For an element $a \in \mathbb{Z}_N$, we define the absolute value

$$|x| := \begin{cases} x & \text{if } x \leq (N-1)/2 \\ -x & \text{otherwise} \end{cases}.$$

We define the group of signed quadratic residues as $\mathbb{QR}_N^+ := \{|x| : x \in \mathbb{QR}_N\}$. We have that $(\mathbb{QR}_N^+, \circ)$ is a cyclic group with order $|\mathbb{QR}_N^+| = \varphi(N)/4$, where, for all $a, b \in \mathbb{QR}_N^+$ and $x \in \mathbb{Z}_N$, group operations are defined as follows:

$$a \circ b := |a \cdot b \bmod N|, \quad a^x := \underbrace{a \circ a \circ \ldots \circ a}_{x \text{ times}} = |a^x \bmod N|, \quad a^{-1} := |a^{-1} \bmod N|.$$

Lemma 1 (Lemma 7, [37]). *Let* $N' := \lceil N/4 \rceil, \mathbb{G} := \mathbb{QR}_N^+$*, and* $X \xleftarrow{\$} \mathbb{Z}_{N'}, Y \xleftarrow{\$} \mathbb{Z}_{|\mathbb{G}|}$*. Then the statistical distance* $D(X, Y)$ *satisfies* $D(X, Y) \leq \frac{2(P+Q)}{PQ}$*.*

2.3 Digital Signature

Definition 3 (Syntax of Digital Signature). *A digital signature scheme* SIG *is a tuple of algorithms* $(\mathsf{Setup}, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ *where*

- *The setup algorithm* Setup *takes as input a security parameter* 1^n*, and outputs system parameters* par*.*
- *The key generation algorithm* Gen *takes as input the system parameters* par*, and returns public and secret keys* (pk, sk)*. We assume that* pk *defines a message space* $\mathcal{M}$ *and a signature space* Σ*.*
- *The signing algorithm* Sign *takes the secret key* sk *and a message* $m \in \mathcal{M}$ *as inputs, and returns a signature* $\sigma \in \Sigma$*.*
- *The* deterministic *verification algorithm* Ver *takes a public key* pk*, a message* m *and a signature* σ *as inputs and returns 1 (accept) or 0 (reject).*

For correctness, we require that $\Pr[\mathsf{Ver}(pk, m, \mathsf{Sign}(sk, m)) = 1] = 1$*.*

Definition 4 (MU-UF-CMA Security). *A signature scheme* SIG *is said to be* $(t, \varepsilon, \ell, Q_s)$-MU-UF-CMA *secure (multi-user unforgeable against chosen message attacks), if for all adversaries* $\mathcal{A}$ *that run in time* t *and makes at most* Q_s *queries to the signature oracle in the security game in Figure 1, we have*

$$\Pr\left[\mathsf{MU\text{-}UF\text{-}CMA}^{\mathcal{A}} \Rightarrow 1\right] \leq \varepsilon.$$

Oracle INITIALIZE:	**Oracle** SIGN(i, m):
$\mathsf{par} \xleftarrow{\$} \mathsf{Setup}(1^n)$	$M \leftarrow M \cup \{(i, m)\}$
For $i = 1, \ldots, \ell$: $(pk_i, sk_i) \xleftarrow{\$} \mathsf{Gen}(\mathsf{par})$	$\mathsf{ctr} := \mathsf{ctr} + 1$
$M := \emptyset$; $\mathsf{ctr} := 0$	$\sigma \leftarrow \mathsf{Sign}(sk_i, m)$
Return $(\mathsf{par}, pk_1, \ldots, pk_\ell)$	Return σ
Oracle FINALIZE(i^*, m^*, σ^*):	
Return $((\mathsf{ctr} \leq Q_s) \wedge \mathsf{Ver}(pk_{i^*}, m^*, \sigma^*) \wedge (i^*, m^*) \notin M)$	

Fig. 1. Security game for MU-UF-CMA security with ℓ users.

3 Construction from the CDH Assumption

Our construction here is based on the CDH-based online/offline signature scheme in [37]. We apply the key-prefixing technique [12] on it.

Let $H_1 \colon \{0,1\}^* \to \mathbb{G}$ and $H_2 \colon \{0,1\}^* \to \{0,1\}^n$ be two hash functions. We recall the signature scheme $\mathsf{OF}_{\mathsf{CDH}} := (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ from [37] and define our key-prefixing variant $\mathsf{PF\text{-}OF}_{\mathsf{CDH}} := (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Sign}_{\mathsf{pf}}, \mathsf{Ver}_{\mathsf{pf}})$ of it in Figure 2. We highlight the differences with grey. By additionally hashing R_1 in H_2, we can prove that the multi-user (MU-UF-CMA) security of our $\mathsf{PF\text{-}OF}_{\mathsf{CDH}}$ can be tightly implied by the single-user security of $\mathsf{OF}_{\mathsf{CDH}}$ in the programmable random oracle model. Interestingly, $\mathsf{PF\text{-}OF}_{\mathsf{CDH}}$ is the same as the original Chevallier-Mames scheme [19]. Our proof can be seen as a new, tight proof of the scheme in the multi-user setting, while the original proof is only tight in the single-user setting.

$\mathsf{Setup}(1^n)$:	$\mathsf{Sign}_{\mathsf{pf}}(sk, m)$:	$\mathsf{Ver}_{\mathsf{pf}}(pk, m, \sigma)$:
$\mathsf{par} := (p, g, \mathbb{G}) \xleftarrow{\$} \mathcal{G}(1^n)$	$r \xleftarrow{\$} \mathbb{Z}_p$; $R_1 := g^r$	Parse $\sigma := (R_L, h_2, s)$
Return par	$h_1 := H_1(R_1)$	$R_1 := g^s \cdot X^{-h_2}$
	$R_L := h_1^x \in \mathbb{G}$; $R_R := h_1^r$	$h_1 := H_1(R_1)$
$\mathsf{Gen}(\mathsf{par})$:	$R_2 := (R_L, R_R)$	$R_R := h_1^s \cdot R_L^{-h_2}$
$sk := x \xleftarrow{\$} \mathbb{Z}_p$	$h_2 := H_2(R_1, R_2, pk, m)$	$R_2 := (R_L, R_R)$
$pk := X = g^x$	$s := x \cdot h_2 + r \in \mathbb{Z}_p$	If $h_2 = H_2(R_1, R_2, pk, m)$
Return (pk, sk)	$\sigma := (R_L, h_2, s)$	Return 1
	Return σ	Else return 0

Fig. 2. Signature schemes $\mathsf{OF}_{\mathsf{CDH}}$ and $\mathsf{PF\text{-}OF}_{\mathsf{CDH}}$. We highlight the difference with grey. Both schemes execute all the codes, while the codes with grey are only executed in $\mathsf{PF\text{-}OF}_{\mathsf{CDH}}$.

We recall the security of $\mathsf{OF}_{\mathsf{CDH}}$ from [37].

Lemma 2 (Security of $\mathsf{OF}_{\mathsf{CDH}}$, Theorem 2 of [37]). *If* CDH *is* (t, ε)*-hard w.r.t* $\mathcal{G}$*, then* $\mathsf{OF}_{\mathsf{CDH}}$ *is* $(t', \varepsilon', Q_s, Q_1, Q_2)$*-*UF-CMA *secure in the programmable*

random oracle model, where

$$\varepsilon' \leq \varepsilon + \frac{Q_2+2}{2^n} + \frac{(Q_1+Q_2)Q_s}{2^n} + \frac{1}{2^n}, \quad t' \approx t. \tag{3}$$

where Q_s, Q_1 and Q_2 are upper bounds on the number of signature and hash queries to H_1 and H_2 in the $\mathsf{UF\text{-}CMA}$*-experiment.*

Lemma 3 ($\mathsf{UF\text{-}CMA}$ of $\mathsf{OF_{CDH}} \to \mathsf{MU\text{-}UF\text{-}CMA}$ of $\mathsf{PF\text{-}OF_{CDH}}$). *If* $\mathsf{OF_{CDH}}$ *is* $(t, \varepsilon, Q_s, Q_1, Q_2)$-$\mathsf{UF\text{-}CMA}$ *secure, then* $\mathsf{PF\text{-}OF_{CDH}}$ *is* $(t', \varepsilon', \ell, Q'_s, Q'_1, Q'_2)$-$\mathsf{MU\text{-}UF\text{-}CMA}$ *secure in the programmable random oracle model, where*

$$\varepsilon' \leq \varepsilon + \frac{Q'_2 Q'_s}{2^n}, \quad Q'_s = Q_s, \quad Q'_1 = Q_1 - 1, \quad Q'_2 = Q_2 - 1, \text{ and } t' \approx t. \tag{4}$$

Here Q_s, Q_1 and Q_2 are upper bounds on the number of signature and hash queries to H_1 and H_2 in the $\mathsf{UF\text{-}CMA}$*-experiment. Similarly, Q'_s, Q'_1 and Q'_2 are upper bounds on the number of signature and hash queries to H'_1 and H'_2 in the* $\mathsf{MU\text{-}UF\text{-}CMA}$*-experiment.*

Combining Lemmata 2 and 3, we get the following theorem.

Theorem 1 (Security of $\mathsf{PF\text{-}OF_{CDH}}$). *If* CDH *is (t, ε)-hard with respect to $\mathcal{G}$, then* $\mathsf{PF\text{-}OF_{CDH}}$ *is* $(t', \varepsilon', \ell, Q_s, Q_1, Q_2)$-$\mathsf{MU\text{-}UF\text{-}CMA}$ *secure in the programmable random oracle model, where*

$$\varepsilon' \leq \varepsilon + \frac{Q_2+3}{2^n} + \frac{(Q_1+Q_2+2)Q_s}{2^n} + \frac{1}{2^n} + \frac{Q_2 Q_s}{2^n}, \quad t' \approx t. \tag{5}$$

where Q_s, Q_1 and Q_2 are upper bounds on the number of signature and hash queries to H_1 and H_2 in the $\mathsf{MU\text{-}UF\text{-}CMA}$*-experiment.*

Thus, we only need to prove Lemma 3.

3.1 Proof of Lemma 3

Let $\mathcal{A}$ be an adversary that $(t', \varepsilon', \ell, Q'_s, Q'_1, Q'_2)$-breaks the $\mathsf{MU\text{-}UF\text{-}CMA}$ security of $\mathsf{PF\text{-}OF_{CDH}}$. We prove Lemma 3 by constructing a reduction $\mathcal{B}$ that $(t, \varepsilon, Q_s, Q_1, Q_2)$-breaks the $\mathsf{UF\text{-}CMA}$ security of $\mathsf{OF_{CDH}}$ and provides oracle access for $\mathcal{A}$ as in Figure 3.

The reduction $\mathcal{B}$ gets oracle access to $\textsc{Initialize}_{\mathsf{U}}$, $\textsc{Sign}_{\mathsf{U}}$, and $\textsc{Finalize}_{\mathsf{U}}$ and random oracles $\textsc{Hash}_1$ and $\textsc{Hash}_2$ (for hash function H_1 and H_2 in $\mathsf{OF_{CDH}}$) from the $\mathsf{UF\text{-}CMA}$ security experiment. Moreover, $\mathcal{B}$ simulates oracles $\textsc{Initialize}_{\mathsf{MU}}$, $\textsc{Sign}_{\mathsf{MU}}$, $\textsc{Finalize}_{\mathsf{MU}}$ and random oracles $\textsc{Hash}'_1$ and $\textsc{Hash}'_2$ (for hash functions H_1 and H_2 in $\mathsf{PF\text{-}OF_{CDH}}$) for adversary $\mathcal{A}$.

Analysis. We show that $\mathcal{B}$ simulates a distribution statistically close to the real one for $\mathcal{A}$. It is trivial to see that the output of $\textsc{Initialize}_{\mathsf{MU}}$ distributes the same as the real one, since X_i is uniformly random over $\mathbb{G}$. Random oracles

Oracle $\textsc{Initialize}_{\mathsf{MU}}$:
$(\mathsf{par}, X) \leftarrow \textsc{Initialize}_{\mathsf{U}}$
For $i := 1, \ldots, \ell$:
  $a_i \xleftarrow{\$} \mathbb{Z}_p$
  $X_i := X \cdot g^{a_i}$
$M := \emptyset$; $\mathsf{ctr} := 0$
Return $(\mathsf{par}, X_1, \ldots, X_\ell)$

Oracle $\textsc{Sign}_{\mathsf{MU}}(i, m)$:
$M \leftarrow M \cup \{(i, m)\}$
$\mathsf{ctr} := \mathsf{ctr} + 1$
$\hat{\sigma} := (\hat{R}_L, \hat{h}_2, \hat{s}) \leftarrow \textsc{Sign}_{\mathsf{U}}(X_i, m)$
$\hat{R}_1 := g^{\hat{s}} \cdot X^{-\hat{h}_2}$
$\hat{h}_1 \leftarrow \textsc{Hash}_1(\hat{R}_1)$
$R_L := \hat{R}_L \cdot \hat{h}_1^{a_i}$; $\hat{R}_R := \hat{h}_1^{\hat{s}} \cdot \hat{R}_L^{-\hat{h}_2}$
$R_2 := (R_L, \hat{R}_R)$
If $\mathrm{H}_2'[\hat{R}_1, R_2, X_i, m] = \bot$ then
  $\mathrm{H}_2'[\hat{R}_1, R_2, X_i, m] := \hat{h}_2$
Else
  Abort
$s := \hat{s} + a_i \hat{h}_2$
Return $\sigma := (R_L, \hat{h}_2, s)$

Oracle $\textsc{Hash}_1'(R)$:
Return $\textsc{Hash}_1(R)$

Oracle $\textsc{Hash}_2'(R_1, R_2, X_j, m)$:
If $X_j = X_i$ for some $1 \le i \le \ell$
  Parse $R_2 := (R_L, R_R)$
  $h_1 \leftarrow \textsc{Hash}_1(R_1)$
  $h_2 \leftarrow \textsc{Hash}_2(R_1, (R_L / h_1^{a_i}, R_R), X_j, m)$
Else
  $h_2 \leftarrow \textsc{Hash}_2(R_1, R_2, X_j, m)$
$\mathrm{H}_2'[R_1, R_2, X_j, m] := h_2$
Return h_2

Oracle $\textsc{Finalize}_{\mathsf{MU}}(i^*, m^*, \sigma^*)$:
If $(i^*, m^*) \in M \wedge \mathsf{ctr} > Q_s$
  Abort
Parse $\sigma^* := (R_L^*, h_2^*, s^*)$
$R_1^* := g^{s^*} \cdot X_{i^*}^{-h_2^*}$
$h_1^* \leftarrow \textsc{Hash}_1'(R_1^*)$
$\tilde{R}_L := R_L^* / {h_1^*}^{a_{i^*}}$
$\tilde{s} := s^* - a_{i^*} h_2^*$
$\tilde{\sigma} := (\tilde{R}_L, h_2^*, \tilde{s})$
Return $\textsc{Finalize}_{\mathsf{U}}((X_{i^*}, m^*), \tilde{\sigma})$

Fig. 3. Security reduction $\mathcal{B}$ to break the UF-CMA security of $\mathsf{OF}_{\mathsf{CDH}}$, and simulate oracles for adversary $\mathcal{A}$ against the MU-UF-CMA security of $\mathsf{PF\text{-}OF}_{\mathsf{CDH}}$. H_2' is a list that keeps track of the inputs and outputs of random oracle $\textsc{Hash}_2'$.

$\textsc{Hash}_1$ and $\textsc{Hash}_2$ are provided by the UF-CMA challenger and thus $\textsc{Hash}_1'$ and $\textsc{Hash}_2'$ are simulated properly.

Our focus is to show that signatures simulated by $\textsc{Sign}_{\mathsf{MU}}$ are statistically close to those outputted by $\mathsf{Sign}_{\mathsf{pf}}$ of $\mathsf{PF\text{-}OF}_{\mathsf{CDH}}$. Given $\hat{\sigma} := (\hat{R}_L, \hat{h}_2, \hat{s}) \leftarrow \textsc{Sign}_{\mathsf{U}}(X_i, m)$, $\hat{\sigma}$ is a valid signature w.r.t. the verification of $\mathsf{OF}_{\mathsf{CDH}}$ (defined in Figure 2) and $\hat{s}$ distributes uniformly at random, namely, the following equation holds:

$$\hat{h}_2 = \textsc{Hash}_2(\hat{R}_2, X_i, m),$$

where $\hat{R}_2 = (\hat{R}_L, \hat{R}_R)$, $\hat{R}_R = \hat{h}_1^{\hat{s}} \cdot \hat{R}_L^{-\hat{h}_2}$, $\hat{h}_1 = \textsc{Hash}_1(\hat{R}_1)$ and $\hat{R}_1 = g^{\hat{s}} \cdot X^{-\hat{h}_2}$.

If $\textsc{Sign}_{\mathsf{MU}}(i, m)$ does not abort, the signature $\sigma := (R_L, \hat{h}_2, s)$ with $s = \hat{s} + a_i \hat{h}_2$ output by $\textsc{Sign}_{\mathsf{MU}}(i, m)$ has the right distribution, namely, s is uniformly random (which is trivial due to the random $\hat{s}$) and σ will pass the verification $\mathsf{Ver}_{\mathsf{pf}}$ of $\mathsf{PF\text{-}OF}_{\mathsf{CDH}}$: Firstly, $\mathsf{Ver}_{\mathsf{pf}}$ will compute values R_1 and $R_2 := (R_L, R_R)$ according to its definition in Figure 2, and, by our simulation of $\textsc{Initialize}_{\mathsf{MU}}$ and $\textsc{Sign}_{\mathsf{MU}}$, the following holds

$$R_1 := g^s \cdot X_i^{-\hat{h}_2} = g^{\hat{s} + a_i \hat{h}_2} \cdot (X \cdot g^{a_i})^{-\hat{h}_2} = g^{\hat{s}} \cdot X^{-\hat{h}_2} = \hat{R}_1$$
$$R_R := h_1^s \cdot R_L^{-\hat{h}_2} = \hat{h}_1^{\hat{s} + a_i \hat{h}_2} \cdot (\hat{R}_L \cdot \hat{h}_1^{a_i})^{-\hat{h}_2} = \hat{h}_1^{\hat{s}} \cdot \hat{R}_L^{-\hat{h}_2} = \hat{R}_R$$

where $h_1 = \textsc{Hash}_1(R_1) = \textsc{Hash}_1(\hat{R}_1) = \hat{h}_1$. Thus, $\hat{h}_2$ in σ returned by $\textsc{Sign}_{\mathsf{MU}}(i, m)$ will have $\hat{h}_2 = \textsc{Hash}'_2(R_1, R_2, X_i, m)$ and $\mathsf{Ver}_{\mathsf{pf}}(X_i, m, \sigma) = 1$.

Moreover, since $\hat{s}$ is uniform, $\hat{R}_1$ distributes uniformly over $\mathbb{G}$ and the probability that $\mathrm{H}'_2[\hat{R}_1, R_2, X_i, m]$ has been defined is at most $Q'_2/|\mathbb{G}|$. By applying the union bound on the number of signing queries, $\mathcal{B}$ will abort its simulation with probability at most $Q_s Q'_2/|\mathbb{G}|$.

A Valid Forgery. To see that $\mathcal{B}$ produces a valid forgery, we first assume that the forgery $(i^*, m^*, \sigma^* = (R^*_L, h^*_2, s^*))$ made by $\mathcal{A}$ is a valid forgery in the MU-UF-CMA-experiment under the public key X_{i^*}, meaning that for

$$R^*_1 := g^{s^*} \cdot X_{i^*}^{-h^*_2}, \quad h^*_1 := \textsc{Hash}_1(R^*_1) \quad \text{and} \quad R^*_R := {h^*_1}^{s^*} \cdot {R^*_L}^{-h^*_2},$$

we have $h^*_2 = \textsc{Hash}'_2(R^*_1, R^*_L, R^*_R, X_{i^*}, m^*)$. In addition, it satisfies the freshness condition that (i^*, m^*) has not been queried in a previous signature query. For the signature $\tilde{\sigma} = (\tilde{R}_L, h^*_2, \tilde{s})$, we compute

$$\tilde{R}_1 := g^{\tilde{s}} \cdot X^{-h^*_2} = g^{s^* - a_{i^*} h^*_2} \cdot X^{-h^*_2} = g^{s^*} \cdot X_{i^*}^{-h^*_2} = R^*_1.$$

We set $\tilde{h}_1 := \textsc{Hash}_1(\tilde{R}_1) = \textsc{Hash}_1(R^*_1) = h^*_1$ and compute

$$\tilde{R}_R := \tilde{h}_1^{\tilde{s}} \cdot \tilde{R}_L^{-h^*_2} = \tilde{h}_1^{s^* - a_{i^*} h^*_2} \cdot \left(R^*_L / \tilde{h}_1^{a_{i^*}}\right)^{-h^*_2} = \tilde{h}_1^{s^*} \cdot {R^*_L}^{-h^*_2} = R^*_R.$$

Then, by the simulation of $\textsc{Hash}'_2(R^*_1, R^*_2, X_{i^*}, m^*)$ and $\tilde{R}_L = R^*_L / h_1^{a_{i^*}}$, we have that

$$\begin{aligned} h^*_2 &= \textsc{Hash}'_2(R^*_1, R^*_2, X_{i^*}, m^*) = \textsc{Hash}'_2(R^*_1, (R^*_L, R^*_R), X_{i^*}, m^*) \\ &= \textsc{Hash}_2(R^*_1, (R^*_L / h_1^{a_{i^*}}, R^*_R), X_{i^*}, m^*) = \textsc{Hash}_2(\tilde{R}_1, (\tilde{R}_L, \tilde{R}_R), X_{i^*}, m^*) = \tilde{h}_2 \end{aligned}$$

and hence $\mathsf{Ver}(X, \tilde{m}, \tilde{\sigma}) = 1$ where $\tilde{m} := (X_{i^*}, m^*)$. Since σ^* was a fresh signature on (i^*, m^*), $\tilde{m}$ has never been queried to the UF-CMA signature oracle, and hence $\tilde{\sigma}$ is a fresh signature on the message $\tilde{m}$.

4 Construction from the Factoring Assumption

We can also apply our method to $\mathsf{FS}_{\mathsf{FAC}}$ in [37] to get tight MU-UF-CMA security from the FAC assumption. We refer readers to Section 2.2 for necessary mathematical background of this section.

Let $H_1 \colon \{0,1\}^* \to \mathbb{QR}^+_N$ and $H_2 \colon \{0,1\}^* \to \{0, \ldots, 2^k - 1\}$ be hash functions, and let g be a generator of $\mathbb{QR}^+_N$. As before, in Figure 4 we have the original scheme $\mathsf{FS}_{\mathsf{FAC}}$ and its prefixed variant $\mathsf{PF\text{-}FS}_{\mathsf{FAC}}$. To give a syntactically correct definition, we require that Setup outputs a private parameter sp that only inputs to Gen.

By combining Corollary 1, Lemma 8[5] and Lemma 1 of [37], we get the following result.

[5] We use the result derived in the reduction, not the statement of the lemma, as they are not the same.

$\underline{\mathsf{Setup}(1^n)}$:	$\underline{\mathsf{Sign}_{\boxed{\mathsf{pf}}}(sk, m)}$:	$\underline{\mathsf{Ver}_{\boxed{\mathsf{pf}}}(pk, m, \sigma)}$:
$p, q \xleftarrow{\$} \mathbb{P}_{n/2}$ s.t.	$r \xleftarrow{\$} \mathbb{Z}_{N/4}; R_1 := g^r$	Parse $\sigma := (R_L, h_2, s)$
$P := 2p + 1 \in \mathbb{P}_{n/2}$	$h_1 := H_1(R_1, \boxed{pk}, m) \in \mathbb{QR}_N^+;$	$R_1 := g^s \circ X^{-h_2}$
$Q := 2q + 1 \in \mathbb{P}_{n/2}$	$R_L := h_1^x; R_R := h_1^r$	$h_1 := H_1(R_1, \boxed{pk}, m)$
$N := PQ$	$R_2 := (R_L, R_R)$	$R_R := h_1^s \circ R_L^{-h_2}$
$\mathsf{par} := (N, g)$	$h_2 := H_2(\boxed{R_1}, R_2, \boxed{pk}, m)$	$R_2 := (R_L, R_R)$
$\mathsf{sp} := (p, q)$	$s :=$	If $h_2 = H_2(\boxed{R_1}, R_2, \boxed{pk}, m)$
Return $(\mathsf{par}, \mathsf{sp})$	$\quad x \cdot h_2 + r \mod (\varphi(N)/4)$	$\quad$ Return 1
	$\sigma := (R_L, h_2, s)$	Else return 0
$\underline{\mathsf{Gen}(\mathsf{par}, \mathsf{sp})}$:	Return σ	
$x \xleftarrow{\$} \mathbb{Z}_{N/4}; X := g^x$		
$sk := (x, p, q)$		
$pk := X$		
Return (pk, sk)		

Fig. 4. Signature schemes $\mathsf{FS}_{\mathsf{FAC}}$ and $\mathsf{PF\text{-}FS}_{\mathsf{FAC}}$. We highlight the difference with $\boxed{\text{grey}}$. Both schemes execute all the codes, while the codes with grey are only executed in $\mathsf{PF\text{-}FS}_{\mathsf{FAC}}$.

Lemma 4 (Security of $\mathsf{FS}_{\mathsf{FAC}}$). *If* FAC *is* (t, ε)*-hard for* Blum_n*, then* $\mathsf{FS}_{\mathsf{FAC}}$ *is* $(t', \varepsilon', Q_1, Q_2)$-UF-KOA *secure in the random oracle model, where*

$$\varepsilon' \leq \varepsilon + \frac{1}{2^{n/2}} + \frac{Q_2 + 1}{2^k}, \quad t' \approx t.$$

Lemma 5 (UF-KOA of $\mathsf{FS}_{\mathsf{FAC}} \to$ MU-UF-CMA of $\mathsf{PF\text{-}FS}_{\mathsf{FAC}}$). *If* $\mathsf{FS}_{\mathsf{FAC}}$ *is* $(t, \varepsilon, Q_1, Q_2)$-UF-KOA *secure, then* $\mathsf{PF\text{-}FS}_{\mathsf{FAC}}$ *is* $(t', \varepsilon', \ell, Q_s, Q_1', Q_2')$-MU-UF-CMA *secure in the programmable random oracle model, where*

$$\varepsilon' \leq \varepsilon + \frac{1}{2^{n/2-2}} + Q_s \left(\frac{Q_1'}{2^n} + \frac{Q_2'}{2^k} \right), \quad Q_1' = Q_1 - 1,\ Q_2' = Q_2 - 1,\ \text{and } t' \approx t. \quad (6)$$

Here Q_1 *and* Q_2 *are upper bounds on the number of hash queries to* H_1 *and* H_2 *in the* UF-KOA*-experiment. Similarly,* Q_s, Q_1' *and* Q_2' *are upper bounds on the number of signature and hash queries to* H_1' *and* H_2' *in the* MU-UF-CMA*-experiment.*

Combining Lemmata 4 and 5, we get the following theorem.

Theorem 2 (Security of $\mathsf{PF\text{-}FS}_{\mathsf{FAC}}$). *If* FAC *is* (t, ε)*-hard for* Blum_n*, then* $\mathsf{PF\text{-}FS}_{\mathsf{FAC}}$ *is* $(t', \varepsilon', Q_s, Q_1, Q_2)$-MU-UF-CMA *secure in the programmable random oracle model, where*

$$\varepsilon' \leq \varepsilon + \frac{1}{2^{n/2}} + \frac{Q_2 + 2}{2^k} + \frac{1}{2^{n/2-2}} + Q_s \left(\frac{Q_1}{2^n} + \frac{Q_2}{2^k} \right), \ t' \approx t. \quad (7)$$

As before, we now only need to prove Lemma 5.

Oracle $\textsc{Initialize}_{\mathsf{MU}}$:
$(\mathsf{par}, X) \leftarrow \textsc{Initialize}_{\mathsf{U}}$
For $i := 1, \ldots, \ell$
  $a_i \xleftarrow{\$} \mathbb{Z}_{\lceil N/4 \rceil}$
  $X_i := X \circ g^{a_i}$
  $pk_i := X_i$
$M := \emptyset$; $\mathsf{ctr} := 0$
Return $(\mathsf{par}, pk_1, \ldots, pk_\ell)$

Oracle $\textsc{Sign}_{\mathsf{MU}}(i, m)$:
$M \leftarrow M \cup \{(i, m)\}$
$\mathsf{ctr} := \mathsf{ctr} + 1$
$\hat{s}, w, \hat{h}_2 \xleftarrow{\$} \mathbb{Z}_{\lceil N/4 \rceil}$
$\hat{h}_1 := g^w$; $\hat{R}_1 := g^{\hat{s}} \circ X_i^{-\hat{h}_2}$
$\hat{R}_L := X_i^w$; $\hat{R}_R := \hat{R}_1^w$
If $\mathrm{H}'_1[\hat{R}_1, X_i, m] = \perp$ then
  $\mathrm{H}'_1[\hat{R}_1, X_i, m] := h_1$
Else
  Abort
$\hat{R}_2 = (\hat{R}_L, \hat{R}_R)$
If $\mathrm{H}'_2[\hat{R}_1, \hat{R}_2, X_i, m] = \perp$ then
  $\mathrm{H}'_2[\hat{R}_1, \hat{R}_2, X_i, m] := \hat{h}_2$
Else
  Abort
Return $\sigma := (\hat{R}_L, \hat{h}_2, \hat{s})$

Oracle $\textsc{Hash}'_1(R, X_j, m)$:
If $\mathrm{H}'_1[R, X_j, m] \neq \perp$
  Return $\mathrm{H}'_1[R, X_j, m]$
$h_1 \leftarrow \textsc{Hash}_1(R, X_j, m)$
$\mathrm{H}'_1[R, X_j, m] := h_1$
Return h_1

Oracle $\textsc{Hash}'_2(R_1, R_2, X_j, m)$:
If $\mathrm{H}'_2[R_1, R_2, X_j, m] \neq \perp$
  Return $\mathrm{H}'_2[R_1, R_2, X_j, m]$
If $X_j = X_i$ for some $1 \leq i \leq \ell$
  Parse $R_2 := (R_L, R_R)$
  $h_1 \leftarrow \textsc{Hash}'_1(R_1, X_i, m)$
  $h_2 \leftarrow \textsc{Hash}_2(R_1, (R_L \circ h_1^{-a_i}, R_R), X_i, m)$
Else
  $h_2 \leftarrow \textsc{Hash}_2(R_1, R_2, X_j, m)$
$\mathrm{H}'_2[R_1, R_2, X_j, m] := h_2$
Return h_2

Oracle $\textsc{Finalize}_{\mathsf{MU}}(i^*, m^*, \sigma^*)$:
If $((i^*, m^*) \in M \wedge \mathsf{ctr} > Q_s)$
  Abort
Parse $\sigma^* := (R_L^*, h_2^*, s^*)$
$R_1^* := g^{s^*} \circ X_{i^*}^{-h_2^*}$
$h_1^* \leftarrow \textsc{Hash}'_1(R_1^*, X_{i^*}, m^*)$
$\tilde{R}_L := R_L^* \circ (h_1^*)^{-a_{i^*}}$
$\tilde{s} := s^* - a_{i^*} h_2^*$
$\tilde{\sigma} := (\tilde{R}_L, h_2^*, \tilde{s})$
Return $\textsc{Finalize}_{\mathsf{U}}((X_{i^*}, m^*), \tilde{\sigma})$

Fig. 5. Security reduction $\mathcal{B}$ to break the UF-KOA security of $\mathsf{FS}_{\mathsf{FAC}}$, and simulate oracles for adversary $\mathcal{A}$ against the MU-UF-CMA security of $\mathsf{PF\text{-}FS}_{\mathsf{FAC}}$. Operations denoted with $\circ$ are performed in $\mathbb{QR}_N^+$, while other operations are performed over the integers.

4.1 Proof of Lemma 5

Let $\mathcal{A}$ be an adversary that breaks the $(t', \varepsilon', \ell, Q_s, Q'_1, Q'_2)$-MU-UF-CMA-security of $\mathsf{PF\text{-}FS}_{\mathsf{FAC}}$. We construct a reduction $\mathcal{B}$ that breaks the $(t, \varepsilon, Q_1, Q_2)$-UF-KOA-security of $\mathsf{FS}_{\mathsf{FAC}}$ as in Figure 5. As before, the reduction $\mathcal{B}$ gets oracle access to $\textsc{Initialize}_{\mathsf{U}}, \textsc{Finalize}_{\mathsf{U}}$ and random oracles $\textsc{Hash}_1$ and $\textsc{Hash}_2$ (for hash function H_1 and H_2 in $\mathsf{FS}_{\mathsf{FAC}}$) from the UF-KOA security experiment. Moreover, $\mathcal{B}$ simulates oracles $\textsc{Initialize}_{\mathsf{MU}}, \textsc{Sign}_{\mathsf{MU}}, \textsc{Finalize}_{\mathsf{MU}}$ and random oracles $\textsc{Hash}'_1$ and $\textsc{Hash}'_2$ (for hash functions H_1 and H_2 in $\mathsf{PF\text{-}FS}_{\mathsf{FAC}}$) for adversary $\mathcal{A}$.

Analysis. We again want to show that $\mathcal{B}$ simulates a distribution statistically close to the real one for $\mathcal{A}$. It is trivial to see that the output of $\textsc{Initialize}_{\mathsf{MU}}$ has the same distribution as in the real case, since X_i is uniformly random over $\mathbb{QR}_N^+$.

The random oracles are provided by the UF-KOA challenger and thus $\textsc{Hash}'_1$ and $\textsc{Hash}'_2$ are properly simulated.

If $\textsc{Sign}_{\mathsf{MU}}$ does not abort, the signature $\sigma = (\hat{R}_L, \hat{h}_2, \hat{s})$ is within statistical distance $2(P+Q)/PQ \leq 2^{2-n/2}$ from a real distribution, and it passes the verification $\mathsf{Ver}_{\mathsf{pf}}$ of $\mathsf{PF\text{-}FS}_{\mathsf{FAC}}$. To show this, we use Lemma 1 and a result from Lemma 8 in [37]. Combined, these show that when simulating a signature like we do in $\textsc{Sign}_{\mathsf{MU}}$, the returned transcript $(\hat{R}_1, \hat{h}_1, \hat{R}_2, \hat{h}_2, \hat{s})$ is within statistical distance at most $2(P+Q)/PQ$ from a real distribution. This is so because $\hat{s}$ has statistical distance at most $2(P+Q)/PQ$ from a uniformly random variable over $\mathbb{Z}_{|\mathbb{QR}_N^+|}$ by Lemma 1, and $\hat{R}_1, \hat{R}_L, \hat{R}_R$ are determined by $\hat{s}, \hat{h}_2$ and X_i, since they are the unique values that satisfy $\hat{R}_1 = g^{\hat{s}} \circ X_i^{-\hat{h}_2}$ and $\hat{R}_R = \hat{h}_1^{\hat{s}} \circ \hat{R}_L^{-\hat{h}_2}$.

For the verification, we proceed as we did for $\mathsf{PF\text{-}OF}_{\mathsf{CDH}}$. The $\mathsf{Ver}_{\mathsf{pf}}$ algorithm computes R_1 and R_R as described in Figure 4, and from our simulation of $\textsc{Initialize}_{\mathsf{MU}}$ and $\textsc{Sign}_{\mathsf{MU}}$ we get

$$R_1 := g^{\hat{s}} \circ X_i^{-\hat{h}_2} = \hat{R}_1$$
$$R_R := h_1^{\hat{s}} \circ \hat{R}_L^{-\hat{h}_2} = g^{w\hat{s}} \circ X_i^{-w\hat{h}_2} = \left(g^{\hat{s}} \circ X_i^{-\hat{h}_2}\right)^w = \hat{R}_1^w = \hat{R}_R,$$

where we after the programming have $h_1 := \textsc{Hash}_1(R_1, X_i, m) = \hat{h}_1 = g^w$. Thus, $\hat{h}_2$ in σ returned by $\textsc{Sign}_{\mathsf{MU}}(i, m)$ will satisfy

$$\hat{h}_2 := \textsc{Hash}'_2(R_1, \hat{R}_L, R_R, X_i, m),$$

and therefore $\mathsf{Ver}_{\mathsf{pf}}(X_i, m, \sigma) = 1$. In the simulation we randomly choose $\hat{s} \xleftarrow{\$} \mathbb{Z}_{\lceil N/4 \rceil}$, which means that R_1 will be uniformly random over $\mathbb{QR}_N^+$, and the probability that $\mathrm{H}'_1[\hat{R}_1, X_i, m]$ has been defined is at most $Q'_1/\left|\mathbb{QR}_N^+\right| \leq Q'_1/2^n$. A similar argument shows that the probability that $\mathrm{H}'_2[\hat{R}_1, R_2, X_i, m]$ has been defined is at most $Q'_2/2^k$. The union bound applied on the number of signing queries shows that $\mathcal{B}$ will abort its simulation with probability at most $Q_s(Q'_1/2^n + Q'_2/2^k)$.

A VALID FORGERY. To show that $(X_{i^*}, m^*, \tilde{\sigma} = (\tilde{R}_L, h_2^*, \tilde{s}))$ is a valid forgery in the UF-KOA-experiment, we first assume that $(i^*, m^*, \sigma^* := (R_L^*, h_2^*, s^*))$ is a valid signature in the MU-UF-CMA-experiment, meaning that for

$$R_1^* := g^{s^*} \circ X_{i^*}^{-h_2^*}, \quad h_1^* := \textsc{Hash}'_1(R_1^*, X_{i^*}, m^*) \quad \text{and} \quad R_R^* := {h_1^*}^{s^*} \circ {R_L^*}^{-h_2^*},$$

we have $h_2^* = \textsc{Hash}'_2(R_1^*, R_L^*, R_R^*, X_{i^*}, m^*)$. It also satisfies the freshness condition that (i^*, m^*) has not been queried in a previous signature query in the MU-UF-CMA game. This means that if $h_1^* = \mathrm{H}'_1[R_1^*, X_{i^*}, m^*]$ or $h_2^* = \mathrm{H}'_2[R_1^*, R_2^*, X_{i^*}, m^*]$ are defined, it was not done by $\textsc{Sign}_{\mathsf{MU}}$, and hence the value was returned by an UF-KOA hash oracle, as required. For the signature $\tilde{\sigma} = (\tilde{R}_L, h_2^*, \tilde{s})$ generated in $\textsc{Finalize}_{\mathsf{MU}}$, we compute $\tilde{R}_1 := g^{\tilde{s}} \circ X^{-h_2^*} = g^{s^* - a_{i^*} h_2^*} \circ X^{-h_2^*} = g^{s^*} \circ X_{i^*}^{-h_2^*} = R_1^*$. We set $\tilde{h}_1 = \textsc{Hash}_1(\tilde{R}_1, X_{i^*}, m^*) =$

$\textsc{Hash}_1(R^*, X_{i^*}, m^*) = h_1^*$, and compute

$$\tilde{R}_R := \tilde{h}_1^{\tilde{s}} \circ \tilde{R}_L^{-h_2^*} = \tilde{h}_1^{s^* - a_{i^*} h_2^*} \circ \left(R_L^* \circ (\tilde{h}_1)^{-a_{i^*}} \right)^{-h_2^*} = \tilde{h}_1^{s^*} \circ {R_L^*}^{-h_2^*} = R_R^*.$$

Then, by the simulation of $\textsc{Hash}'_2(R_1^*, R_2^*, X_{i^*}, m^*)$, we have that

$$\tilde{h}_2 := \textsc{Hash}_2(\tilde{R}_1, (\tilde{R}_L, \tilde{R}_R), X_{i^*}, m^*) = \textsc{Hash}'_2(R_1^*, R_2^*, X_{i^*}, m^*) = h_2^*, \quad (8)$$

and hence $\mathsf{Ver}(X, \tilde{m}, \tilde{\sigma}) = 1$ where $\tilde{m} := (X_{i^*}, m^*)$. The running time is that of $\mathcal{A}$ plus the Q_s simulations of $\textsc{Sign}_{\mathsf{MU}}$, and we write $t' \approx t$.

References

1. Abdalla, M., Fouque, P.A., Lyubashevsky, V., Tibouchi, M.: Tightly-secure signatures from lossy identification schemes. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 572–590. Springer, Heidelberg (Apr 2012) 5
2. Abe, M., Jutla, C.S., Ohkubo, M., Pan, J., Roy, A., Wang, Y.: Shorter QA-NIZK and SPS with tighter security. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part III. LNCS, vol. 11923, pp. 669–699. Springer, Heidelberg (Dec 2019) 2
3. Abe, M., Jutla, C.S., Ohkubo, M., Roy, A.: Improved (almost) tightly-secure simulation-sound QA-NIZK with applications. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018, Part I. LNCS, vol. 11272, pp. 627–656. Springer, Heidelberg (Dec 2018) 2
4. Bader, C., Hofheinz, D., Jager, T., Kiltz, E., Li, Y.: Tightly-secure authenticated key exchange. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part I. LNCS, vol. 9014, pp. 629–658. Springer, Heidelberg (Mar 2015) 2, 3
5. Bader, C., Jager, T., Li, Y., Schäge, S.: On the impossibility of tight cryptographic reductions. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, Part II. LNCS, vol. 9666, pp. 273–304. Springer, Heidelberg (May 2016) 7
6. Bellare, M., Poettering, B., Stebila, D.: From identification to signatures, tightly: A framework and generic transforms. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part II. LNCS, vol. 10032, pp. 435–464. Springer, Heidelberg (Dec 2016) 3
7. Bellare, M., Ristenpart, T.: Simulation without the artificial abort: Simplified proof and improved concrete security for Waters' IBE scheme. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 407–424. Springer, Heidelberg (Apr 2009) 2
8. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Denning, D.E., Pyle, R., Ganesan, R., Sandhu, R.S., Ashby, V. (eds.) ACM CCS 93. pp. 62–73. ACM Press (Nov 1993) 3
9. Bellare, M., Rogaway, P.: The exact security of digital signatures: How to sign with RSA and Rabin. In: Maurer, U.M. (ed.) EUROCRYPT'96. LNCS, vol. 1070, pp. 399–416. Springer, Heidelberg (May 1996) 5
10. Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 409–426. Springer, Heidelberg (May / Jun 2006) 8

11. Bernstein, D.J.: Proving tight security for Rabin-Williams signatures. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 70–87. Springer, Heidelberg (Apr 2008) 3
12. Bernstein, D.J.: Multi-user Schnorr security, revisited. Cryptology ePrint Archive, Report 2015/996 (2015), http://eprint.iacr.org/2015/996 8, 10, 20
13. Blazy, O., Kakvi, S.A., Kiltz, E., Pan, J.: Tightly-secure signatures from chameleon hash functions. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 256–279. Springer, Heidelberg (Mar / Apr 2015) 2, 3
14. Blazy, O., Kiltz, E., Pan, J.: (Hierarchical) identity-based encryption from affine message authentication. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 408–425. Springer, Heidelberg (Aug 2014) 2, 3, 8
15. Blocki, J., Lee, S.: On the multi-user security of short schnorr signatures. Cryptology ePrint Archive, Report 2019/1105 (2019), https://eprint.iacr.org/2019/1105 4
16. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited (preliminary version). In: 30th ACM STOC. pp. 209–218. ACM Press (May 1998) 4
17. Chatterjee, S., Koblitz, N., Menezes, A., Sarkar, P.: Another look at tightness II: Practical issues in cryptography. Cryptology ePrint Archive, Report 2016/360 (2016), http://eprint.iacr.org/2016/360 2
18. Chen, J., Wee, H.: Fully, (almost) tightly secure IBE and dual system groups. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 435–460. Springer, Heidelberg (Aug 2013) 2, 3
19. Chevallier-Mames, B.: An efficient CDH-based signature scheme with a tight security reduction. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 511–526. Springer, Heidelberg (Aug 2005) 3, 4, 10
20. Cohn-Gordon, K., Cremers, C., Gjøsteen, K., Jacobsen, H., Jager, T.: Highly efficient key exchange protocols with optimal tightness. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part III. LNCS, vol. 11694, pp. 767–797. Springer, Heidelberg (Aug 2019) 2, 7
21. den Boer, B.: Diffie-Hellman is as strong as discrete log for certain primes (rump session). In: Goldwasser, S. (ed.) CRYPTO'88. LNCS, vol. 403, pp. 530–539. Springer, Heidelberg (Aug 1990) 6
22. Fersch, M., Kiltz, E., Poettering, B.: On the provable security of (EC)DSA signatures. In: Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S. (eds.) ACM CCS 2016. pp. 1651–1662. ACM Press (Oct 2016) 4
23. Galbraith, S., Paterson, K., Smart, N.: Pairings for cryptographers. Cryptology ePrint Archive, Report 2006/165 (2006), http://eprint.iacr.org/2006/165 6
24. Galbraith, S.D., Malone-Lee, J., Smart, N.P.: Public key signatures in the multi-user setting. Inf. Process. Lett. 83(5), 263–266 (2002), http://dx.doi.org/10.1016/S0020-0190(01)00338-6 2, 3, 4
25. Gay, R., Hofheinz, D., Kiltz, E., Wee, H.: Tightly CCA-secure encryption without pairings. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, Part I. LNCS, vol. 9665, pp. 1–27. Springer, Heidelberg (May 2016) 2
26. Gay, R., Hofheinz, D., Kohl, L., Pan, J.: More efficient (almost) tightly secure structure-preserving signatures. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part II. LNCS, vol. 10821, pp. 230–258. Springer, Heidelberg (Apr / May 2018) 2, 3
27. Gjøsteen, K., Jager, T.: Practical and tightly-secure digital signatures and authenticated key exchange. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 95–125. Springer, Heidelberg (Aug 2018) 2, 3, 5, 6, 7

28. Goh, E.J., Jarecki, S., Katz, J., Wang, N.: Efficient signature schemes with tight reductions to the Diffie-Hellman problems. Journal of Cryptology 20(4), 493–514 (Oct 2007) 2, 3, 4, 5, 6
29. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. Comput. 17(2), 281–308 (1988), https://doi.org/10.1137/0217017 2
30. Guo, F., Chen, R., Susilo, W., Lai, J., Yang, G., Mu, Y.: Optimal security reductions for unique signatures: Bypassing impossibilities with a counterexample. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 517–547. Springer, Heidelberg (Aug 2017) 2, 3
31. Hofheinz, D., Jager, T.: Tightly secure signatures and public-key encryption. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 590–607. Springer, Heidelberg (Aug 2012) 2, 3
32. Hofheinz, D., Kiltz, E.: Programmable hash functions and their applications. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 21–38. Springer, Heidelberg (Aug 2008) 8
33. Kakvi, S.A., Kiltz, E.: Optimal security proofs for full domain hash, revisited. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 537–553. Springer, Heidelberg (Apr 2012) 5
34. Kakvi, S.A., Kiltz, E.: Optimal security proofs for full domain hash, revisited. Journal of Cryptology 31(1), 276–306 (Jan 2018) 3
35. Katz, J., Wang, N.: Efficiency improvements for signature schemes with tight security reductions. In: Jajodia, S., Atluri, V., Jaeger, T. (eds.) ACM CCS 2003. pp. 155–164. ACM Press (Oct 2003) 3
36. Kerry, C.F., Director, C.R.: Fips pub 186-4 federal information processing standards publication digital signature standard (dss) (2013) 4, 5, 6, 20
37. Kiltz, E., Loss, J., Pan, J.: Tightly-secure signatures from five-move identification protocols. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part III. LNCS, vol. 10626, pp. 68–94. Springer, Heidelberg (Dec 2017) 3, 4, 5, 6, 7, 8, 9, 10, 13, 16
38. Kiltz, E., Masny, D., Pan, J.: Optimal security proofs for signatures from identification schemes. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part II. LNCS, vol. 9815, pp. 33–61. Springer, Heidelberg (Aug 2016) 3, 4, 5, 6, 7, 20
39. Maurer, U.M., Wolf, S.: Diffie-Hellman oracles. In: Koblitz, N. (ed.) CRYPTO'96. LNCS, vol. 1109, pp. 268–282. Springer, Heidelberg (Aug 1996) 6
40. Micali, S., Reyzin, L.: Improving the exact security of digital signature schemes. Journal of Cryptology 15(1), 1–18 (Jan 2002) 3, 5
41. Morgan, A., Pass, R.: On the security loss of unique signatures. In: Beimel, A., Dziembowski, S. (eds.) TCC 2018, Part I. LNCS, vol. 11239, pp. 507–536. Springer, Heidelberg (Nov 2018) 7
42. Schnorr, C.P.: Efficient signature generation by smart cards. Journal of Cryptology 4(3), 161–174 (Jan 1991) 5, 6
43. Wu, G., Lai, J., Guo, F., Susilo, W., Zhang, F.: Tightly secure public-key cryptographic schemes from one-more assumptions. J. Comput. Sci. Technol. 34(6), 1366–1379 (2019), https://doi.org/10.1007/s11390-019-1980-2 3, 5, 6
44. Zhang, X., Liu, S., Gu, D., Liu, J.K.: A generic construction of tightly secure signatures in the multi-user setting. Theor. Comput. Sci. 775, 32–52 (2019), https://doi.org/10.1016/j.tcs.2018.12.012 3

A On the Multi-User Security of DSA

We show why it is difficult to show tight implication from the single-user security to the multi-user security for DSA. We first recall the scheme. Let p be an L-bit prime, and q be an N-bit prime such that $q \mid (p-1)$. For specifications on L and N, see the DSA documentation [36]. Let g be a generator of a subgroup of order q in $\mathbb{Z}_p^*$. The $\mathsf{Gen}, \mathsf{Sign}$ and Ver can then be described as follows.

$\underline{\mathsf{Gen}(\mathsf{par})}$:	$\underline{\mathsf{Sign}(sk, m)}$:	$\underline{\mathsf{Ver}(X, m, \sigma)}$:
$sk := x \xleftarrow{\$} \mathbb{Z}_q$	$r \xleftarrow{\$} \mathbb{Z}_q^*$	Parse $\sigma := (R, s)$
$X := g^x \mod p$	$R := (g^r \mod p) \mod q$	If $R = 0 \vee s = 0$
$pk := X$	$s :=$	Return 0
Return (pk, sk)	$\left(r^{-1}(H(m) + xR)\right) \mod q$	$w := s^{-1} \mod q$
	Return $\sigma := (R, s)$	$u_1 := H(m) \cdot w \mod q$
		$u_2 := R \cdot w \mod q$
		$v :=$
		$(g^{u_1} X^{u_2} \mod p) \mod q$
		If $v = R$
		Return 1
		Else return 0

Different to the Schnorr signature, given a valid signature $\sigma := (R, s)$ under public key X, it is not possible to convert it to a valid signature under public key $X \cdot g^{a_i}$ for $a_i \xleftarrow{\$} \mathbb{Z}_q^* = \mathbb{Z}_q \setminus \{0\}$ using methods in [38,12], since we do not have the discrete log of R, namely, $r \in \mathbb{Z}_q^*$.