Optimal Security Proofs for Signatures from Identification Schemes

Eike Kiltz Daniel Masny Jiaxin Pan

Horst-Görtz Institute for IT Security and Faculty of Mathematics,
Ruhr-University Bochum, Germany
{eike.kiltz, daniel.masny, jiaxin.pan}@rub.de

Abstract

We perform a concrete security treatment of digital signature schemes obtained from canonical identification schemes via the Fiat-Shamir transform. If the identification scheme is random self-reducible and satisfies the weakest possible security notion (key-recoverability), then the signature scheme obtained via Fiat-Shamir is unforgeable against chosen-message attacks in the multi-user setting. Our security reduction is in the random oracle model and loses a factor of roughly Q_h, the number of hash queries. Previous reductions incorporated an additional multiplicative loss of N, the number of users in the system. Our analysis is done in small steps via intermediate security notions, and all our implications have relatively simple proofs. Furthermore, for each step, we show the optimality of the given reduction in terms of model assumptions and tightness.

As an important application of our framework, we obtain a concrete security treatment for Schnorr signatures.

Keywords: Signatures, Identification, Schnorr, tightness

1 Introduction

Canonical Identification Schemes and the Fiat-Shamir Transform. A canonical identification scheme ID as formalized by Abdalla et al. [AABN02] is a three-move public-key authentication protocol of a specific form. The prover (holding the secret-key) sends a commitment R to the verifier. The verifier (holding the public-key) returns a random challenge h, uniformly chosen from a set ChSet (of exponential size). The prover sends a response s. Finally, using the verification algorithm, the verifier publicly checks correctness of the transcript (R, h, s). There is a large number of canonical identification schemes known (e.g. [FS87, GQ90, Bet88, MS90, Sch91, BM91, Gir91, OS91, Oka93, KW03, GJKW07], the most popular among them being the scheme by Schnorr [Sch91]. The Fiat-Shamir method [FS87] transforms any such canonical identification scheme into a digital signature scheme SIG[ID] using a hash function.

Digital Signatures in the Multi-User Setting. When it comes to security of digital signature schemes, in the literature almost exclusively the standard security notion of unforgeability against chosen message attacks (UF-CMA) [GMR88] is considered. This is a *single-user setting*, where an adversary obtains one single public-key and it is said to break the scheme's security if he can produce (after obtaining Q_s many signatures on messages of his choice) a valid forgery, i.e. a message-signature pair that verifies on the given public-key. However, in the real world the attacker is usually confronted with many public-keys and presumably he is happy if he can produce a valid forgery under any of the given public-keys. This scenario is captured in the *multi-user setting* for signatures schemes. Concretely, in multi-user unforgeability against chosen message attacks (MU-UF-CMA) the attacker obtains N independent public-keys and is said to break the scheme's security if he can produce (after obtaining Q_s many signatures on public-keys of his choice) a valid forgery that verifies under any of the public-keys.

There are essentially two reasons why one typically only analyzes signatures in the single-user setting. First, the single-user security notion and consequently their analysis are simpler. Second, there exists a simple generic security reduction [GMS02] between multi-user security and standard single-user security. Namely, for any signature system, attacking the scheme in the multi-user setting with N public-keys cannot decrease the attacker's success ratio (i.e., the quotient of its success probability and its running time) by a factor more than N compared to attacking the scheme in the single-user setting. As the number of public-keys N is bounded by a polynomial, asymptotically, the single-user and the multi-user setting are equivalent. However, the security reduction is not tight: it has a loss of a non-constant factor

N. This is clearly not satisfactory as in complex environments one can easily assume the existence of at least $N = 2^{30}$ public-keys, thereby increasing the upper bound on the attacker's success ratio by a factor of 2^{30}. For example, if we assume the best algorithm breaking the single-user security having success ratio $\rho = 2^{-80}$, then it can only be argued that the best algorithm breaking the multi-user security has success ratio $\rho' = 2^{-80} \cdot 2^{30} = 2^{-50}$, which is not a safe security margin that defends against today's attackers.

TIGHTNESS. Generally, we call a security implication between two problems *tight*, if the success ratio ρ of any adversary attacking the first problem cannot decease by more than a small constant factor compared to the success ratio ρ' of any adversary attacking the second problem. Here the success ratio ρ is defined as the quotient between the adversary's success probability and its running time [BR09]. We note that this notion of tightness is slightly weaker than requiring that both, success probability and running time, cannot decrease by more than a small constant factor. However, the main goal of a concrete security analysis is to derive parameters provably guaranteeing *k-bit security.* As the term *k-bit security* is commonly defined as the non-existence of any adversary that breaks the scheme with a success ratio better than 2^{-k} (see, e.g., [BR09]), our definition of tightness is sufficient for this purpose.

1.1 Our Contributions

This work contains a concrete and modular security analysis of signatures SIG[ID] obtained via the Fiat-Shamir transform. Throughout this paper we assume that our canonical identification schemes ID are Σ-protocols, i.e. they are honest-verifier zero-knowledge (HVZK), have special soundness (SS), and commitments R are sampled at random from a sufficiently large set. For some of our tight implications we furthermore require ID to be random self-reducible (RSR), a property we formally define in Definition 2.8. Most known canonical identification schemes satisfy the above properties.

SECURITY NOTIONS. For identification schemes we consider XXX-YYY security, where $\mathsf{XXX} \in \{\mathsf{KR}, \mathsf{IMP}, \mathsf{PIMP}\}$ denotes the attacker's goal and $\mathsf{YYY} \in \{\mathsf{KOA}, \mathsf{PA}\}$ the attacker's capabilities. If the attacker's goal is key-recovery (KR), then it tries to compute a valid secret-key; in impersonation (IMP), it tries to impersonate a prover by convincing an honest verifier; parallel impersonation (PIMP) is a parallel version of IMP, where the adversary tries to convince a verifier in one of Q_{CH} many parallel sessions. In a key-only attack (KOA), the adversary is only given the public-key; in a passive attack (PA), the adversary is provided with valid transcripts between an honest prover and verifier. By the above definitions we obtain $3 \times 2 = 6$ different security notions that that were all previously considered in the literature [PS00, OO98, AABN02], except PIMP-YYY security.

OVERVIEW. We show via a chain of implications that KR-KOA-security (the weakest possible security notion for ID where the adversary has to compute a secret-key from a given public-key) implies multi-user unforgeability against chosen message attacks (MU-UF-CMA) of SIG[ID]. All our implications are optimal in terms of tightness and model requirements in the following sense. If one implication makes use of a special model requirement, we prove its impossibility without this requirement. For example, our implication PIMP-KOA → UF-KOA requires the random oracle model [BR93] (with its well-known deficiencies [CGH98]) and we show that the non-programmable random oracle model [FLR$^+$10] is not sufficient to prove the same implication. Exactly one of our implications, namely IMP-KOA → PIMP-KOA is non-tight, and we prove the impossibility of such a tight implication. The diagram in Figure 1 summarizes our results. We now discuss them in more detail.

FROM IDENTIFICATION TO SINGLE USER SECURITY FOR SIGNATURES. Our first main theorem can be informally stated as follows.

Theorem 1.1. *If the identification scheme is* KR-KOA*-secure against any adversary having success ratio ρ, then* SIG[ID] *is* UF-CMA*-secure in the random oracle model against any adversary having success ratio $\rho' \approx \rho/Q_h$, where Q_h is the maximal number of the adversary's random oracle queries.*

The proof of this theorem is done in four independent Lemmas 3.4, 3.5, 3.5, and 3.7 via intermediate security notions IMP-KOA, PIMP-KOA, and UF-KOA[1] security, see Figure 1. We certainly do not claim any novelty of the above lemmas, nor a new proof technique. For example, the implication IMP-KOA → UF-CMA is already explicitly contained in [OO98] (and implicitly in the seminal paper by

[1] Unforgeability against key-only attack (UF-KOA security) is the same as standard UF-CMA security, but the adversary is not allowed to ask any signing query.

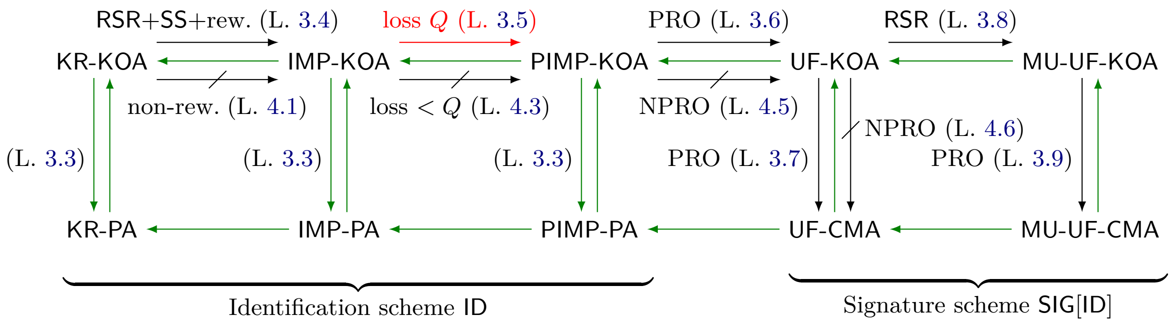


Figure 1: Overview of our notions and results for canonical identification schemes ID and their implied signature schemes SIG[ID]. X $\xrightarrow{Z}$ Y means that X-security implies Y-security under condition Z. Trivial implications are denoted with green arrows. All implications are tight except the one marked with red. The conditions are: rew. (reduction rewinds), loss Q (reduction loses a factor of Q), PRO (reduction is in the programmable random oracle model), SS (reduction uses special soundness), and RSR (reduction uses random self-reducibility for tightness). All implications from top to bottom require HVZK. X $\overset{Z}{\not\rightarrow}$ Y means that X-security does not imply Y-security unless they fulfill condition Z. The conditions are: non-rew. (reduction does not rewind), loss $< Q$ (reduction loses a factor smaller than Q), and NPRO (reduction is in the non-programmable random oracle model).

Pointcheval and Stern [PS00]). However, by our specific choice of the intermediate security notions, all four proofs are extremely simple and intuitive. We remark that, unlike previous proofs, none of our proofs requires the full power of the Forking Lemma [PS00]. Lemma 3.4 (KR-KOA → PIMP-KOA) is the only proof using rewinding and its analysis contains a simple application of Jensen's inequality. We view identifying the intermediate security notions that allow for simple proofs as a conceptual contribution. Our result show that IMP-KOA and PIMP-KOA security can be seen as the *tightness barrier* for identification schemes in the sense that PIMP-KOA is the weakest of our notions for ID that is tightly equivalent to MU-UF-CMA security of SIG[ID] in the random oracle model, whereas IMP-KOA is tightly equivalent to KR-KOA.

One particular advantage of our modular approach is that we are able to prove optimality of all four implications via meta-reductions (Lemmas 4.1, 4.3, 4.5, and 4.6). Lemma 4.3 proving the impossibility of a tight reduction between PIMP-KOA and IMP-KOA security is a generalization of Seurin's impossibility result to canonical identification schemes [Seu12]; Lemmas 4.5 and 4.6 proving the impossibility of a reduction in the non-programmable random oracle model between PIMP-KOA, UF-KOA, and UF-CMA can be considered as a fine-grained version of a general impossibility result by Fukumitsu and Hasegawa [FH15] who only consider the implication IMP-PA → UF-CMA; Lemma 4.1 involves a new meta-reduction. All our impossibility results assume the reductions to be key-preserving [PV05] and are conditional in the sense that the existence of a reduction would imply that ID does not satisfy some other natural security property (that is believed to hold).

From Single-User to Multi-User Security for Signatures. Our second main theorem can be informally stated as follows.

Theorem 1.2. *If* ID *is* UF-KOA*-secure against any adversary having success ratio ρ, then it is* MU-UF-CMA*-secure in the random oracle model against any adversary having success ratio $\rho' \approx \rho/4$, independent of the number of users N in the multi-user scenario.*

This theorem improves the bound implied by previous generic reductions [GMS02] by a factor of N. Following our modular approach, the theorem is proved in two steps via Lemmas 3.8 and 3.9. Lemma 3.8 proves that UF-KOA tightly implies MU-UF-KOA. Tightness stems from the RSR property, meaning that from a given public key pk we can derive properly distributed $pk_1, \ldots, pk_N$ such that any signature σ which is valid under pk can be transformed into a signature σ_i which is valid under pk_i and vice-versa.

Lemma 3.9 is our main technical contribution and proves MU-UF-KOA → MU-UF-CMA in the programmable random oracle model, again with a tight reduction. One is tempted to believe that it can be proved the same way as in the single user setting (i.e., the same way as $\mathsf{UF\text{-}KOA} \rightarrow \mathsf{UF\text{-}CMA}$). In the single user setting, the reduction simulates signatures on m_j using the HVZK property to obtain a valid transcript (R_j, h_j, s_j) and programs the random oracle as $H(R_j, m_j) := h_j$. However, in the MU-UF-KOA experiment an adversary can ask for a signature under pk_1 on message m which makes the reduction program the random oracle $H(R_1, m) := h_1$. Now, if the adversary submits a forgery (R_1, s_2) under pk_2 on the same message m, the reduction cannot use this forgery to break the MU-UF-KOA experiment because the random oracle $H(R_1, m)$ was externally defined by the reduction. Hence, for the MU-UF-KOA experiment, $m, (R_1, s_2)$ does not constitute a valid forgery. In order to circumvent the above problem we make a simple probabilistic argument. In our reduction, about one half of the multi-user public-keys are coming from the MU-UF-KOA experiment, for the other half the reduction knows the corresponding secret-keys. Which secret-keys are known is hidden from the adversary's view. Now, if the multi-user adversary first obtains a signature on message m under pk_1 and then submits a forgery on the same message m under pk_2, the reduction hopes for the good case that one of the public-keys comes from the MU-UF-KOA experiment and the other one is known. This happens with probability $1/4$ which is precisely the loss of our new reduction.

1.2 Example Instantiations

Schnorr Signatures. One of the most important and signature schemes in the discrete logarithm setting is the Schnorr signature scheme [Sch91]. It is obtained via the Fiat-Shamir transform applied to the Schnorr identification protocol. The recent expiry of the patent in 2008 has triggered a number of initiatives to obtain standardized versions of it.

Theorems 1.1 and 1.2 can be used to derive a concrete security bound for (strong) multi-user MU-UF-CMA-security of Schnorr signatures in the random oracle model from the DLOG problem.[2] Our reduction loses a factor of roughly Q_h, the number of random oracle queries. This improves previous bounds by a factor of N, the number of users in the system. We derive concrete example parameters for a provably secure instantiation. Figure 1 shows that DLOG is tightly equivalent to IMP-KOA-security and PIMP-KOA-security is tightly equivalent to MU-UF-CMA-security, meaning the tightness barrier for Schnorr lies precisely between IMP-KOA and PIMP-KOA security.

Katz-Wang Signatures. The Katz-Wang identification scheme [KW03, GJKW07] is a double-generator version of Schnorr. It is at least as secure as Schnorr which means one cannot hope for a tight security reduction to the DLOG assumption. However, we can use a simple argument from [KW03] for a tight security proof of its PIMP-KOA security under the Decision Diffie-Hellman Assumption. By our framework, this implies a tight proof of its (strong) MU-UF-CMA-security.

Other Signatures. Other canonical identification schemes of interest with the required properties include the ones by Guillou-Quisquater [GQ88] and Okamoto [Oka93]. Similar to Katz-Wang, for the Guillou-Quisquater scheme, we can use an argument from [ABP13] for a tight proof of PIMP-KOA security under the Phi-hiding assumption. Alternatively, we can give a proof with loss Q_h under the Factoring assumption. Our framework also shows that this loss is unavoidable. For Okamoto's scheme, we can provide the same bounds as for Schnorr.

1.3 Related Work

Single-User Security. There have been many different works addressing the single-user security of Fiat-Shamir based signature schemes SIG[ID]. In pioneering work, Pointcheval and Stern [PS00] introduced the Forking Lemma as a tool to prove UF-CMA security of SIG[ID] from HVZK, SS and KR-KOA-security. Ohta and Okamoto [OO98] gave an alternative proof from IMP-KOA security and HVZK. Abdalla et al. [AABN02] prove the equivalence of IMP-PA-security of ID and UF-CMA security of SIG[ID] in the random oracle model. All above results incorporate a security loss of at least Q_h and can be seen as a special case of our framework. Furthermore, [BP02] consider stronger security notions (e.g., IMP-AA and man-in-the middle security) for the Schnorr and GQ identification schemes. Abdalla et al. [AFLT12] show

[2] We can even prove *strong* MU-UF-CMA security of Schnorr signatures in the sense that a new signature on a previously signed message already counts as a valid forgery.

that lossy identification schemes tightly imply UF-CMA-secure signatures in the random oracle model from decisional assumptions.

Multi-user security. To mitigate the generic security loss problem in the multi-user setting for the special case of Schnorr's signature scheme, Galbraith, Malone-Lee, and Smart (GMLS) proved [GMS02] a tight reduction, namely that attacking the Schnorr signatures in the multi-user setting with N public-keys provably cannot decrease (by more than a small constant factor) the attacker's success ratio compared to attacking the scheme in the single-user setting. Unfortunately, Bernstein [Ber15b] recently pointed out an error in the GMLS proof leaving a tight security reduction for Schnorr signatures as an open problem. Even worse, Bernstein identifies an "apparently insurmountable obstacle to the claimed [GMLS] theorem". Section 4.3 of [Ber15b] further expands on the insurmountable obstacle. Our Theorem 1.2 shows there is such a tight security reduction for Schnorr signatures if one is willing to rely on the random oracle model. Additionally, in Theorem B.1 we also prove an alternative tight reduction in the standard model which assumes *strong* UF-CMA security. (Schnorr is generally believed to be strongly UF-CMA secure and this is provably equivalent to UF-CMA security in the random oracle model.) Proving the original GMLS theorem (i.e., without random oracles and from standard UF-CMA security) remains an open problem.

Impossibility Results. In terms of impossibility results, Seurin [Seu12], building on earlier work of [PV05, GBL08], proves that there is no tight reduction from the (one-more) discrete logarithm assumption to UF-KOA-security of Schnorr signatures. A more recent result by [FJS14] even excludes a reduction from any non-interactive assumption.[3] Fukumitsu and Hasegawa [FH15], generalizing earlier work on Schnorr signatures [FF13, PV05], prove that SIG[ID] cannot be proved secure in the non-programmable random oracle model only assuming IMP-PA security of ID.

Schnorr signatures vs. Key-Prefixed Schnorr signatures. After identifying the error in the GMLS proof, Bernstein [Ber15b] uses the lack of a tight security reduction for Schnorr's signature scheme as a motivation to promote a "key-prefixed" modification to Schnorr's signature scheme which includes the verifier's public-key in the hash function. The EdDSA signature scheme by Bernstein, Duif, Lange, Schwabe, and Yang [BDL$^+$11] is essentially a key-prefixing variant of Schnorr's signature scheme. (In the context of security in a multi-user setting, key-prefixing was considered before, e.g., in [BGLS03].) In [BDL$^+$11] key-prefixing is advertized as "an inexpensive way to alleviate concerns that several public keys could be attacked simultaneously." Indeed, Bernstein [Ber15b] proves that single-user security of the original Schnorr signatures scheme tightly implies multi-user security of the key-prefixed variant of the scheme. That is, the key-prefixed variant has the advantage of a standard model proof of its tight multi-user security, whereas for standard Schnorr signatures one has to assume strong security or rely on the random oracle model.

The TLS standard used to secure HTTPS connections is maintained by the Internet Engineering Task Force (IETF) which delegates research questions to the Internet Research Task Force (IRTF). Cryptographic research questions are usually discussed in the Crypto Forum Research Group (CFRG) mailing list. In the last months the CFRG discussed the issue of key-prefixing.

Key-prefixing comes with the disadvantage that the entire public-key has to be available at the time of signing. Specifically, in a CFRG message from September 2015 Hamburg [Ham15] argues "having to hold the public key along with the private key can be annoying" and "can matter for constrained devices". Independent of efficiency, we believe that a cryptographic protocol should be as light as possible and prefixing (just as any other component) should only be included if its presence is justified. Naturally, in light of the GMLS proof, Hamburg [Ham15] and Struik [Str15] (among others) recommended against key prefixing for Schnorr. Shortly after, Bernstein [Ber15a] identifies the error in the GMLS theorem and posts a tight security proof for the key-prefixed variant of Schnorr signatures. In what happens next, the participant of the CFRG mailing list switched their minds and mutually agree that key-prefixing should be preferred, despite of its previously discussed disadvantages. Specifically, Brown writes about Schnorr signatures that "this justifies a MUST for inclusion of the public key in the message of the classic signature" [Bro15]. As a consequence, key-prefixing is contained in the current draft for EdDSA [JL]. In the light of our new results, we recommend to reconsider this decision.

[3] The main result of the published paper [FJS14] even excludes reduction from any *interactive* assumption (with special algebraic properties), but the proof turned out to be flawed.

2 Definitions

2.1 Preliminaries

For an integer p, define $[p] := \{1, \ldots, p\}$ and $\mathbb{Z}_p$ as the residual ring $\mathbb{Z}/p\mathbb{Z}$. If A is a set, then $a \overset{\$}{\leftarrow} A$ denotes picking a from A according to the uniform distribution. All our algorithms are probabilistic polynomial time unless stated otherwise. If A is an algorithm, then $a \overset{\$}{\leftarrow} \mathsf{A}(b)$ denotes the random variable which is defined as the output of $\mathcal{A}$ on input b. To make the randomness explicit, we use the notation $a := (A)(b; \mathbf{t})$ meaning that the algorithm is executed on input b and randomness $\mathbf{t}$. Note that A's execution is now deterministic.

2.2 Digital Signatures

We now define syntax and security of a digital signature scheme. Let par be common system parameters shared among all participants.

Definition 2.1 (Digital Signature). *A digital signature scheme* SIG *is defined as a triple of algorithms* $\mathsf{SIG} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$.

- *The key generation algorithm* $\mathsf{Gen}(\mathsf{par})$ *returns the public and secret keys* (pk, sk).
- *The signing algorithm* $\mathsf{Sign}(sk, m)$ *returns a signature* σ.
- *The deterministic verification algorithm* $\mathsf{Ver}(pk, m, \sigma)$ *returns 1 (accept) or 0 (reject).*

We require that for all $(pk, sk) \in \mathsf{Gen}(\mathsf{par})$, *all messages* $m \in \{0,1\}^*$, *we have* $\mathsf{Ver}(pk, m, \mathsf{Sign}(sk, m)) = 1$.

Definition 2.2 (Multi-user Security). *A signature scheme* SIG *is said to be* (t, ε, N, Q_s)-$\mathsf{MU\text{-}SUF\text{-}CMA}$ *secure (multi-user strongly unforgeable against chosen message attacks) if for all adversaries* $\mathcal{A}$ *running in time at most* t *and making at most* Q_s *queries to the signing oracle,*

$$\Pr\left[\begin{array}{l|l} \mathsf{Ver}(pk_{i^*}, m^*, \sigma^*) = 1 & \textit{For } i = 1, \ldots, N : (pk_i, sk_i) \overset{\$}{\leftarrow} \mathsf{Gen}(\mathsf{par}) \\ \wedge\ (i^*, m^*, \sigma^*) \notin \{(i_j, m_j, \sigma_j) \mid j \in [Q_s]\} & (i^*, m^*, \sigma^*) \overset{\$}{\leftarrow} \mathcal{A}^{\textsc{Sign}(\cdot,\cdot)}(pk_1, \ldots, pk_N) \end{array}\right] \leq \varepsilon,$$

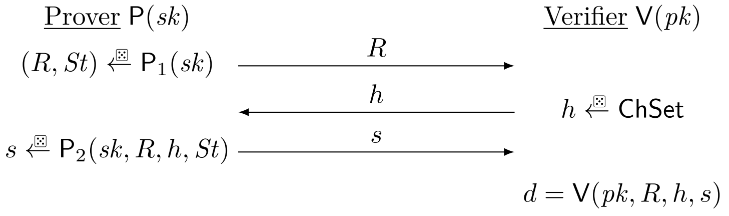
where on the j*-th query* $(i_j, m_j) \in [N] \times \{0,1\}^*$ $(j \in [Q_s])$ *the signing oracle* $\textsc{Sign}$ *returns* $\sigma_j \overset{\$}{\leftarrow} \mathsf{Sign}(sk_{i_j}, m_j)$ *to* $\mathcal{A}$, *i.e., a signature on message* m_j *under public-key* pk_{i_j}.

We stress that an adversary in particular breaks multi-user security if he asks for a signature on message m under pk_1 and submits a valid forgery on the same message m under pk_2.

The first condition in the probability statement of Definition 2.2 is called the correctness condition, the second condition is called the freshness condition. Definition 2.2 covers *strong* security in the sense that a new signature on a previously queried message is considered as a fresh forgery. For standard (non-strong) $\mathsf{MU\text{-}UF\text{-}CMA}$ security (multi-user unforgeablility against chosen message attack) we modify the freshness condition in the experiment to $(i^*, m^*) \notin \{(i_j, m_j,) \mid j \in [Q_s]\}$, i.e., to break the scheme the adversary has to come up with a signature on a message-key pair which has not been queried to the signing oracle. We also define (t, ε, N)-$\mathsf{MU\text{-}UF\text{-}KOA}$ security (multi-user unforgeability against key only attack) as $(t, \varepsilon, N, 0)$-$\mathsf{MU\text{-}UF\text{-}CMA}$ security, i.e. $Q_s = 0$, the adversary is not allowed to make any signing query

Definition 2.3 (Single-user Security). *In the single-user setting, i.e.* $N = 1$ *users,* (t, ε, Q_s)-$\mathsf{SUF\text{-}CMA}$ *security (strong unforgeablility against chosen message attacks) is defined as* $(t, \varepsilon, 1, Q_s)$-$\mathsf{MU\text{-}SUF\text{-}CMA}$ *security. Similarly, standard (non-strong)* (t, ε, Q_s)-$\mathsf{UF\text{-}CMA}$ *security (unforgeablility against chosen message attack) is defined as* $(t, \varepsilon, 1, Q_s)$-$\mathsf{MU\text{-}UF\text{-}CMA}$ *security. Further,* (t, ε)-$\mathsf{UF\text{-}KOA}$ *security (unforgeablility against key-only attack) is defined as* $(t, \varepsilon, 1, 0)$-$\mathsf{MU\text{-}SUF\text{-}CMA}$ *security, i.e.,* $N = 1$ *users and* $Q_s = 0$ *signing queries.*

Security in the random oracle model. The security of signature scheme containing a hash function can be analyzed in the random oracle model [BR93]. In this model hash values can only be accessed by an adversary through queries to an oracle H. On input x this oracle returns a uniformly random output $H(x)$ which is consistent with previous queries for input x. Using the random oracle model, the maximal number of queries to H becomes a parameter in the concrete security notions. For example, for $(t, \varepsilon, N, Q_s, Q_h)$-$\mathsf{MU\text{-}SUF\text{-}CMA}$ security we consider all adversaries making at most Q_h queries to the random oracle. We make the convention that each query to the random oracle made during a signing query is counted as the adversary's random oracle query, meaning $Q_h \geq Q_s$.



Prover $\mathsf{P}(sk)$
Verifier $\mathsf{V}(pk)$
$(R, St) \xleftarrow{\$} \mathsf{P}_1(sk)$
R
h
$h \xleftarrow{\$} \mathsf{ChSet}$
$s \xleftarrow{\$} \mathsf{P}_2(sk, R, h, St)$
s
$d = \mathsf{V}(pk, R, h, s)$

Figure 2: A canonical identification scheme and its transcript (R, h, s).

2.3 Canonical Identification Schemes

A canonical identification scheme ID is a three-move protocol of the form depicted in Figure 2. The prover's first message R is called *commitment*, the verifier selects a uniform *challenge* h from set ChSet, and, upon receiving a *response* s from the prover, makes a deterministic decision.

Definition 2.4 (Canonical Identification Scheme). *A canonical identification scheme* ID *is defined as a tuple of algorithms* $\mathsf{ID} := (\mathsf{IGen}, \mathsf{P}, \mathsf{ChSet}, \mathsf{V})$*.*

- *The key generation algorithm* IGen *takes system parameters* par *as input and returns public and secret key* (pk, sk)*. We assume that* pk *defines* ChSet*, the set of challenges.*
- *The prover algorithm* $\mathsf{P} = (\mathsf{P}_1, \mathsf{P}_2)$ *is split into two algorithms.* P_1 *takes as input the secret key* sk *and returns a commitment* R *and a state* St*;* P_2 *takes as input the secret key* sk*, a commitment* R*, a challenge* h*, and a state* St *and returns a response* s*.*
- *The verifier algorithm* V *takes the public key* pk *and the conversation transcript as input and outputs a* deterministic decision*, 1 (acceptance) or 0 (rejection).*

We require that for all $(pk, sk) \in \mathsf{IGen}(\mathsf{par})$*, all* $(R, St) \in \mathsf{P}_1(sk)$*, all* $h \in \mathsf{ChSet}$ *and all* $s \in \mathsf{P}_2(sk, R, h, St)$*, we have* $\mathsf{V}(pk, R, h, s) = 1$*.*

We make a couple of useful definitions. An identification scheme ID is called *unique* if for all $(pk, sk) \in \mathsf{IGen}(\mathsf{par})$, $(R, St) \in \mathsf{P}_1(sk)$, $h \in \mathsf{ChSet}$, there exists at most one response $s \in \{0,1\}^*$ such that $\mathsf{V}(pk, R, h, s) = 1$. A *transcript* is a three-tuple (R, h, s). It is called *valid* (with respect to public-key pk) if $\mathsf{V}(pk, R, h, s) = 1$. Furthermore, it is called *real*, if it is the output of a real interaction between prover and verifier as depicted in Figure 2. A canonical identification schemes ID has α *bits of min-entropy*, if for all $(pk, sk) \in \mathsf{IGen}(\mathsf{par})$, the commitment generated by the prover algorithm is chosen from a distribution with at least α bits of min-entropy. That is, for all strings R' we have $\Pr[R = R'] \leq 2^{-\alpha}$, if $(R, St) \xleftarrow{\$} \mathsf{P}_1(sk)$ was honestly generated by the prover.

We now define (parallel) impersonation against key-only attack (KOA), passive attack (PA), and active attack (AA).

Definition 2.5 ((Parallel) Impersonation). *Let* $\mathsf{YYY} \in \{\mathsf{KOA}, \mathsf{PA}, \mathsf{AA}\}$*. A canonical identification* ID *is said to be* $(t, \varepsilon, Q_{\textsc{Ch}}, Q_{\mathrm{O}})$-$\mathsf{PIMP}$-$\mathsf{YYY}$ *secure (parallel impersonation against* YYY *attacks) if for all adversaries* $\mathcal{A}$ *running in time at most* t *and making at most* $Q_{\textsc{Ch}}$ *queries to the challenge oracle* $\textsc{Ch}$ *and* Q_{O} *queries to oracle* O*,*

$$\Pr\left[\mathsf{V}(pk, R_{i^*}, h_{i^*}, s_{i^*}) = 1 \wedge i^* \in [Q_{\textsc{Ch}}] \;\middle|\; \begin{array}{l} (pk, sk) \xleftarrow{\$} \mathsf{IGen}(\mathsf{par}) \\ St \xleftarrow{\$} \mathcal{A}^{\mathrm{O}(\cdot)}(pk) \\ (i^*, s_{i^*}) \xleftarrow{\$} \mathcal{A}^{\textsc{Ch}(\cdot)}(pk) \end{array} \right] \leq \varepsilon,$$

where on the i*-th query* $\textsc{Ch}(R_i)$ $(i \in [Q_{\textsc{Ch}}])$*, the challenge oracle returns* $h_i \xleftarrow{\$} \mathsf{ChSet}$ *to* $\mathcal{A}$*.*[4] *Depending on* YYY*, oracle* O *is defined as follows.*

- *If* $\mathsf{YYY} = \mathsf{KOA}$ *(key-only attack), then* O *always returns* $\perp$*.*
- *If* $\mathsf{YYY} = \mathsf{PA}$ *(passive attack), then* $\mathrm{O} := \textsc{Tran}$*, where on the* j*-th empty query* $\textsc{Tran}(\epsilon)$ $(j \in Q_{\mathrm{O}})$*, the transcript oracle returns a real transcript* (R'_j, h'_j, s'_j) *to* $\mathcal{A}$*, where* $(R'_j, St'_j) \xleftarrow{\$} \mathsf{P}_1(sk)$*,* $h'_j \xleftarrow{\$} \mathsf{ChSet}$*;* $s'_j \xleftarrow{\$} \mathsf{P}_2(sk, R'_j, h'_j, St'_j)$*.*

[4] On two queries $\textsc{Ch}(R_i)$ and $\textsc{Ch}(R_{i'})$ with the same input $R_i = R_{i'}$ the oracle returns two independent random challenges $h_i \xleftarrow{\$} \mathsf{ChSet}$ and $h_{i'} \xleftarrow{\$} \mathsf{ChSet}$.

- *If* $\mathsf{YYY} = \mathsf{AA}$ *(active attack), then* $\mathrm{O} := \textsc{Prover} = (\textsc{Prover}_1, \textsc{Prover}_2)$*, where on the* j*-th query* $\textsc{Prover}_1(\epsilon)$ $(j \in Q_{\mathrm{O}})$*, the prover oracle returns* R'_j *for* $(R'_j, St'_j) \overset{\$}{\leftarrow} \mathsf{P}_1(sk)$ *to* $\mathcal{A}$*; on query* $\textsc{Prover}_2(j, h'_j)$*, the oracle returns* $s'_j \overset{\$}{\leftarrow} \mathsf{P}_2(sk, R'_j, h'_j, St'_j)$*, if* R'_j *is already defined (and* $\perp$ *otherwise).*

If $\mathsf{YYY} = \mathsf{KOA}$*, then the parameter* Q_{O} *is not used and we simply speak of* $(t, \varepsilon, Q_{\mathrm{CH}})$-$\mathsf{PIMP}$-$\mathsf{KOA}$*. Moreover,* $(t, \varepsilon, Q_{\mathrm{O}})$-$\mathsf{IMP}$-$\mathsf{YYY}$ *(*impersonation against YYY attack*) security is defined as* $(t, \varepsilon, 1, Q_{\mathrm{O}})$-$\mathsf{PIMP}$-$\mathsf{YYY}$ *security, i.e., the adversary is only allowed* $Q_{\mathrm{CH}} = 1$ *query to the* $\textsc{Ch}$ *oracle.*

Definition 2.6 (Key-recovery). *Let* $\mathsf{YYY} \in \{\mathsf{KOA}, \mathsf{PA}, \mathsf{AA}\}$*. A canonical identification* ID *is said to be* (t, ε)-KR-YYY *secure (key recovery under* YYY *attack) if for all adversaries* $\mathcal{A}$ *running in time at most* t*,*

$$\Pr\left[\ (sk^*, pk) \in \mathsf{IGen}(\mathsf{par}) \ \middle|\ \begin{array}{l} (pk, sk) \overset{\$}{\leftarrow} \mathsf{IGen}(\mathsf{par}) \\ sk^* \overset{\$}{\leftarrow} \mathcal{A}^{\mathrm{O}(\cdot)}(pk) \end{array} \right] \leq \varepsilon,$$

where depending on YYY *oracle* O *is defined as in Definition 2.5. The winning condition* $(sk^*, pk) \in \mathsf{IGen}(\mathsf{par})$ *means that the tuple* (sk^*, pk) *is in the support of* $\mathsf{IGen}(\mathsf{par})$*, i.e., that* $\mathcal{A}$ *outputs a valid secret-key* sk^* *with respect to* pk*.*

Definition 2.7 (Special Soundness). *A canonical identification* ID *is said to be* SS *(special sound) if there there exists an extractor algorithm* Ext *such that, for all* $(pk, sk) \in \mathsf{IGen}(\mathsf{par})$*, given any two accepting transcripts* (R, h, s) *and* (R, h', s') *(where* $h \neq h'$*), we have* $\Pr[(sk^*, pk) \in \mathsf{IGen}(\mathsf{par}) \mid sk^* \overset{\$}{\leftarrow} \mathsf{Ext}(pk, R, h, s, h', s')] = 1$*.*

Definition 2.8 (Random Self-reducibility). *A canonical identification* ID *is said to be* RSR *(random self-reducible) if there is an algorithm* Rerand *and two deterministic algorithms* Tran *and* Derand *such that, for all* $(pk, sk) \in \mathsf{IGen}(\mathsf{par})$*:*

- pk' *and* pk'' *have the same distribution, where* $(pk', \mathbf{a}') \overset{\$}{\leftarrow} \mathsf{Rerand}(pk)$ *is the rerandomized key-pair and* $(pk'', sk'') \overset{\$}{\leftarrow} \mathsf{IGen}(\mathsf{par})$ *is a freshly generated key-pair.*
- *For all* $(pk', \mathbf{a}') \in \mathsf{Rerand}(pk)$*, all* $(pk', sk') \in \mathsf{IGen}(\mathsf{par})$*, and* $sk^* = \mathsf{Derand}(pk, pk', sk', \mathbf{a}')$*, we have* $(pk, sk^*) \in \mathsf{IGen}(\mathsf{par})$*, i.e.,* Derand *returns a valid secret-key* sk^* *with respect to* pk*, given any valid* sk' *for* pk'*.*
- *For all* $(pk', \mathbf{a}') \in \mathsf{Rerand}(pk)$*, all transcripts* (R', h', s') *that are valid with respect to* pk'*, the transcript* $(R', h', s := \mathsf{Tran}(pk, pk', \mathbf{a}', (R', h', s')))$ *is valid with respect to* pk*.*

Definition 2.9 (Honest-verifier Zero-knowledge). *A canonical identification* ID *is said to be (perfect)* HVZK *(honest-verifier zero-knowledge) if there exists an algorithm* Sim *that, given public key* pk*, outputs* (R, h, s) *such that* (R, h, s) *is a real (i.e., properly distributed) transcript with respect to* pk*.*

2.4 Signatures from Identification Schemes

Let $\mathsf{ID} := (\mathsf{IGen}, \mathsf{P}, \mathsf{ChSet}, \mathsf{V})$ be a canonical identification scheme. By the generalized Fiat-Shamir transformation [BP02], the signature scheme $\mathsf{SIG}[\mathsf{ID}] := (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ from ID is defined as follows. par contains the system parameters of ID and a hash function $H : \{0,1\}^* \to \mathsf{ChSet}$.

```
Gen(par):
  $(pk, sk) \overset{\$}{\leftarrow} \mathsf{IGen}(\mathsf{par})$
  Return $(pk, sk)$
```

```
Sign(sk, m):
  $(R, St) \overset{\$}{\leftarrow} \mathsf{P}_1(sk)$
  $h = H(R, m)$
  $s \overset{\$}{\leftarrow} \mathsf{P}_2(sk, R, h, St)$
  Return $\sigma = (R, s)$
```

```
Ver(sk, m, σ):
  Parse $\sigma = (R, s)$
  $h = H(R, m)$
  Return $\mathsf{V}(pk, R, h, s)$
```

In some variants of the Fiat-Shamir transform, the hash additionally inputs some public parameters, for example $h = H(pk, R, m)$.

We call ID *commitment-recoverable*, if $\mathsf{V}(pk, R, h, s)$ first recomputes the commitment via $R' = \mathsf{V}'(pk, h, s)$ and then outputs 1 iff $R' = R$. For commitment-recoverable ID, we can define an alternative Fiat-Shamir transformation $\mathsf{SIG}'[\mathsf{ID}] := (\mathsf{Gen}, \mathsf{Sign}', \mathsf{Ver}')$. Algorithm $\mathsf{Sign}'(sk, m)$ is defined as $\mathsf{Sign}(sk, m)$, but outputs $\sigma' = (h, s)$. Algorithm $\mathsf{Ver}'(pk, m, \sigma')$ first parses $\sigma' = (h, s)$, then recomputes $R' := \mathsf{V}'(pk, h, s)$, and finally returns 1 iff $H(R', m) = h$. Since $\sigma = (R, s)$ can publicly transformed into $\sigma' = (h, s)$ and vice-cersa, $\mathsf{SIG}[\mathsf{ID}]$ and $\mathsf{SIG}[\mathsf{ID}']$ are equivalent in terms of security. On the one hand, the alternative Fiat-Shamir transform yields shorter signatures if $h \in \mathsf{ChSet}$ has a smaller representation size than response s. On the other hand, signatures of the Fiat-Shamir transform maintain their algebraic structure which in some cases enables useful properties such as batch verification.

3 Security Implications

In this section we will prove the following two main results.

Theorem 3.1 (Main Theorem 1). *Suppose* ID *is* SS*,* HVZK*,* RSR *and has α bit min-entropy. If* ID *is* (t, ε)-KR-KOA *secure then* SIG[ID] *is* $(t', \varepsilon', Q_s, Q_h)$-UF-CMA-*secure and* $(t'', \varepsilon'', N, Q_s, Q_h)$-MU-UF-CMA-*secure in the programmable random oracle model, where*

$$\begin{aligned} \frac{\varepsilon'}{t'} &\leq 6(Q_h + 1) \cdot \frac{\varepsilon}{t} + \frac{Q_s}{2^\alpha} + \frac{1}{|\mathsf{ChSet}|}, \\ \frac{\varepsilon''}{t''} &\leq 24(Q_h + 1) \cdot \frac{\varepsilon}{t} + \frac{Q_s}{2^\alpha} + \frac{1}{|\mathsf{ChSet}|}, \end{aligned}$$

The proof of Theorem 3.1 is obtained by combining Lemmas 3.4-3.9 and using $Q_h \leq t' - 1$.

Theorem 3.2 (Main Theorem 2). *Suppose* SIG[ID] *is* HVZK*,* RSR *and has α bit min-entropy. If* SIG[ID] *is* $(t, \varepsilon, Q_h + Q_s)$-UF-KOA *secure then* SIG[ID] *is* $(t', \varepsilon', N, Q_s, Q_h)$-MU-UF-CMA *secure in the programmable random oracle model, where*

$$\varepsilon' \leq 4\varepsilon + \frac{Q_h Q_s}{2^\alpha}, \qquad t' \approx t$$

and Q_s, Q_h are upper bounds on the number of signing and hash queries in the MU-UF-CMA *experiment, respectively.*

The proof of Theorem 3.2 is obtained by combining Lemmas 3.8 and 3.9.

3.1 Proof of the Main Theorems

Lemma 3.3 (XXX-KOA $\rightarrow$ XXX-PA)**.** *Let* XXX $\in$ {KR, IMP, PIMP}*. If* ID *is* $(t, \varepsilon, Q_{\text{CH}})$-XXX-KOA *secure and* HVZK*, then* ID *is* $(\approx t, \varepsilon, Q_{\text{CH}}, Q_{\text{O}})$-XXX-KOA *secure.*

Proof. Let $\mathcal{A}$ be an adversary against the $(t, \varepsilon, Q_{\text{CH}}, Q_{\text{O}})$-XXX-KOA-security of ID. We now build an adversary $\mathcal{B}$ against the $(t, \varepsilon, Q_{\text{O}})$-XXX-KR security of ID, with (t, ε) as claimed.

Construction of $\mathcal{B}$. Adversary $\mathcal{B}$ inputs pk and runs $\mathcal{A}$ on pk. Essentially, $\mathcal{B}$ only has to simulate the Tran oracle of the passive attack PA in the first phase. All queries to the Ch oracle (for YYY $\in$ {IMP, PIMP}) in the second phase are echoed by $\mathcal{B}$ to its own Ch oracle. Finally, $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs. A query to the Tran oracle can be perfectly simulated by computing a simulated proof via $(R, h, s) \overset{\$}{\leftarrow} \mathsf{Sim}(pk)$, and returning a real transcript (R, h, s). The running time of $\mathcal{B}$ is that of $\mathcal{A}$ plus roughly Q_{O} executions of Sim to simulate the Prover oracle, which we ignore for simplicity. ■

Lemma 3.4 below proving that KR-KOA tightly implies IMP-KOA can be viewed as a generalization of Bellare and Palacio's Reset Lemma [BP02] that takes advantage of ID's random self-reducibility (RSR). Compared to PIMP-KOA, the IMP-KOA security experiment is relatively simple in the sense that the adversary makes exactly one query to the challenge oracle Ch. Therefore the reduction in the proof of the lemma does not have to guess which of the Q_{CH} many challenges the adversary is using to break security. This is the reason why its proof is considerably simpler than the corresponding previous proofs analyzing the security of identification/signature schemes using rewinding, for example the Forking Lemma [PS00, BN06] or the proofs in [Seu12, OO98, MR02].

Lemma 3.4 (KR-KOA $\xrightarrow{\textbf{rewinding}}$ IMP-KOA)**.** *If* ID *is* (t, ε)-KR-KOA *secure,* SS *and* RSR*, then* ID *is* (t', ε')-IMP-KOA *secure, where for any $N > 0$,*

$$\varepsilon \geq (1 - (1 - \varepsilon' + \frac{1}{|\mathsf{ChSet}|})^N)^2, \quad t \approx 2Nt'. \tag{1}$$

In particular, the two success ratios are related as

$$\frac{\varepsilon'}{t'} \leq 6 \cdot \frac{\varepsilon}{t} + \frac{1}{t'|\mathsf{ChSet}|}. \tag{2}$$

For $N = 1$ this is essentially the Reset Lemma [BP02] and the proof does not require RSR. I.e., without RSR, we we can still obtain the weaker bounds $\varepsilon \geq \varepsilon'(\varepsilon' - \frac{1}{|\mathsf{ChSet}|})$, $t \approx 2t'$.

Proof. We first show how to derive (2) from (1). If $\varepsilon' \leq 1/|\mathsf{ChSet}|$, then (2) holds trivially. Assuming $\varepsilon' > 1/|\mathsf{ChSet}|$, we set $N := (\varepsilon' - 1/|\mathsf{ChSet}|)^{-1}$ to obtain $t \approx 2t'/(\varepsilon' - 1/|\mathsf{ChSet}|)$ and $\varepsilon \geq (1 - \frac{1}{e})^2 \geq \frac{1}{3}$. Dividing ε' by t' yields (2).

To prove (1), let $\mathcal{A}$ be an adversary against the (t', ε')-IMP-KOA-security of ID. We now build an adversary $\mathcal{B}$ against the (t, ε)-KR-KOA security of ID, with (t, ε) as claimed in (1).

Construction of $\mathcal{B}$. In phase 1, for each $i \in [N]$, $\mathcal{B}$ does the following. it picks random tape $\mathbf{t}_i$, runs $(pk_i, \mathbf{a}_i) \xleftarrow{\$} \mathsf{Rerand}(pk)$ and executes $\mathcal{A}(pk_i; \mathbf{t}_i)$. On query R_i, $\mathcal{B}$ answers with $h_i \xleftarrow{\$} \mathsf{ChSet}$ to obtain s_i from $\mathcal{A}$. If any of $\mathcal{A}$'s executions produces a valid transcript, i.e., if there exists an index $i^* \in [N]$ such that transcript $(R_{i^*}, h_{i^*}, s_{i^*})$ is a valid transcript with respect to pk_{i^*}, then $\mathcal{B}$ continues its execution. Otherwise, it aborts.

In phase 2, $\mathcal{B}$ fixes i^* and, for each $j \in [N]$, it does the following. It executes $\mathcal{A}(pk_{i^*}; \mathbf{t}_{i^*})$. Adversary $\mathcal{A}$ will always query R_{i^*}, which $\mathcal{B}$ answers with $h'_j \xleftarrow{\$} \mathsf{ChSet} \setminus \{h_{i^*}\}$ to obtain s'_j from $\mathcal{A}$. If any of $\mathcal{A}$'s executions produces a valid transcript, i.e., if there exists an index $j^* \in [N]$ such that transcript $(R_{i^*}, h'_{j^*}, s'_{j^*})$ is a valid transcript with respect to pk_{i^*}, then $\mathcal{B}$ continues its execution. Otherwise, it aborts.

Finally, $\mathcal{B}$ uses the SS property of ID and computes $sk_{i^*} \leftarrow \mathsf{Ext}(pk_{i^*}, R_{i^*}, h_{i^*}, s_{i^*}, h'_{j^*}, s'_{j^*})$. By the RSR property of ID, it returns $sk = \mathsf{Derand}(pk_{i^*}, sk_{i^*}, \mathbf{a}_{i^*})$ and terminates.

Success Probability of $\mathcal{B}$. For each $i \in [N]$, pk_i is a properly distributed public-key and

$$\Pr[\mathsf{V}(pk_i, R_i, h_i, s_i) = 1] = \varepsilon'.$$

Therefore,

$$\Pr[\text{no abort in phase 1}] = 1 - (1 - \varepsilon')^N. \tag{3}$$

Next, for each $i, j \in [N]$ and fixed pk_i and random coins $\mathbf{t}_i$, we define

$$\begin{aligned} q_{i,j} = q_{i,j}(pk_i, \mathbf{t}_i) &:= \Pr_{h_i}[\mathsf{V}(pk_i, R_i, h_i, s_i) = 1], \\ p_{i,j} = p_{i,j}(pk_i, \mathbf{t}_i) &:= \Pr_{h_i, h'_j}[\mathsf{V}(pk_i, R_i, h_i, s_i) = 1 \wedge \mathsf{V}(pk_i, R_i, h'_j, s'_j) = 1], \end{aligned}$$

Note that the value R_i deterministically depends on pk_i and $\mathbf{t}_i$, the value s_i deterministically depends on $pk_i, \mathbf{t}_i$, and h_i, and the value s'_j deterministically depends on $pk_i, \mathbf{t}_i$, and h'_j. Therefore, the probability-space of $p_{i,j}$ is the collection of random variables (h_i, h'_j), where $h_i \xleftarrow{\$} \mathsf{ChSet}$ and $h'_j \xleftarrow{\$} \mathsf{ChSet} \setminus \{h_i\}$. Since $\Pr[h_i = h'_j] = \frac{1}{|\mathsf{ChSet}|}$, we have

$$p_{i,j} \geq q_{i,j} \cdot \left(q_{i,j} - \frac{1}{|\mathsf{ChSet}|} \right).$$

We bound the expectation of $p_{i,j}$ as

$$\begin{aligned} \mathsf{E}_{\mathbf{t}_i, pk_i}[p_{i,j}] &\geq \mathsf{E}_{\mathbf{t}_i, pk_i}\left[q_{i,j} \cdot \left(q_{i,j} - \frac{1}{|\mathsf{ChSet}|} \right) \right] \\ &\geq \mathsf{E}_{\mathbf{t}_i, pk_i}[q_{i,j}] \cdot \left(\mathsf{E}_{\mathbf{t}_i, pk_i}[q_{i,j}] - \frac{1}{|\mathsf{ChSet}|} \right) \\ &= \varepsilon' \left(\varepsilon' - \frac{1}{|\mathsf{ChSet}|} \right). \end{aligned}$$

In the last inequation we used Jensen's inequality[5] applied to the convex function $\varphi(q_{i,j}) := q_{i,j}(q_{i,j} - 1/|\mathsf{ChSet}|)$, where $1/|\mathsf{ChSet}|$ is a constant. Using the above bound, we analyze the probability $\varepsilon_{i,j}$ that

[5] Jensen's inequality states that if φ is a convex function and X is a random variable, then $\mathsf{E}[\varphi(X)] \geq \varphi(\mathsf{E}[X])$.

transcript (R_i, h'_j, s'_j) is valid with respect to pk_i conditioned on the event that transcript (R_i, h_i, s_i) is valid with respect to pk_i.

$$\begin{aligned} \varepsilon_{i,j} &= \Pr[\mathsf{V}(pk_i, R_i, h'_j, s'_j) = 1 \mid \mathsf{V}(pk_i, R_i, h_i, s_i) = 1] \\ &= \frac{\Pr[\mathsf{V}(pk_i, R_i, h'_j, s'_j) = 1 \wedge \mathsf{V}(pk_{i^*}, R_i, h_i, s_i) = 1]}{\Pr[\mathsf{V}(pk_i, R_i, h_i, s_i) = 1]} \\ &= \frac{\mathsf{E}_{\mathbf{t}_i, pk_i}[p_{i,j}]}{\varepsilon'} \geq \varepsilon' - \frac{1}{|\mathsf{ChSet}|}. \end{aligned}$$

Finally, we obtain

$$\Pr[\text{no abort in phase 2} \mid \text{no abort in phase 1}] = 1 - (1 - \varepsilon_{i^*,j})^N = 1 - (1 - \varepsilon' + \frac{1}{|\mathsf{ChSet}|})^N. \qquad (4)$$

As $\mathcal{B}$ is successful if it does not abort, by (3), (4) we obtain

$$\varepsilon \geq (1 - (1 - \varepsilon' + \frac{1}{|\mathsf{ChSet}|})^N)^2.$$

The running time t of $\mathcal{B}$ is $2Nt'$ plus the N times the time to run the Rerand and Derand algorithms of RSR plus the time to run the Ext algorithm of SS. We write $t' \approx 2Nt'$ to indicate that this is the dominating running time of $\mathcal{B}$. ■

Lemma 3.5 (IMP-KOA $\xrightarrow{\text{loss } Q}$ PIMP-KOA). *If* ID *is* (t, ε)-IMP-KOA *secure, then* ID *is* $(t', \varepsilon', Q_{\text{CH}})$-PIMP-KOA *secure, where*

$$\varepsilon' \leq Q_{\text{CH}} \cdot \varepsilon,\ t' \approx t.$$

Proof. Let $\mathcal{A}$ be an adversary against the $(t', \varepsilon', Q_{\text{CH}})$-PIMP-KOA-security of ID. We now build an adversary $\mathcal{B}$ against the (t, ε)-IMP-KOA security of ID, with (t, ε) as claimed.

Construction of $\mathcal{B}$. First, $\mathcal{B}$ obtains pk from its IMP-KOA experiment and forwards it to $\mathcal{A}$. Next, it picks $i^* \xleftarrow{\$} [Q_{\text{CH}}]$. On $\mathcal{A}$'s i-th query $\text{CH}_{\mathcal{A}}(R_i)$, it proceeds as follows. If $i \neq i^*$, then it returns $h_i \xleftarrow{\$} \mathsf{ChSet}$. If $i = i^*$, then it defines $R := R_{i^*}$, makes a query $h \xleftarrow{\$} \text{CH}_{\mathcal{B}}(R)$ to its own challenge oracle, and returns $h_{i^*} := h$ to $\mathcal{A}$. Eventually, $\mathcal{A}$ submits (i, s) and terminates. If $i \neq i^*$, then $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ outputs s to its own experiment and terminates. Clearly, if $i = i^*$ then $\mathcal{B}$ wins if $\mathcal{A}$ wins. Since i^* is uniform in $[Q_{\text{CH}}]$ the probability that this happens is $1/Q_{\text{CH}}$. ■

Lemma 3.6 (PIMP-KOA $\xrightarrow{\textbf{PRO}}$ UF-KOA). *If* ID *is* $(t, \varepsilon, Q_{\text{CH}})$-PIMP-KOA *secure, then* SIG[ID] *is* (t', ε', Q_h)-UF-KOA *secure in the programmable random oracle model, where*

$$\varepsilon' = \varepsilon,\ t' \approx t,\ Q_h = Q_{\text{CH}} - 1.$$

Proof. Let $\mathcal{A}$ be an adversary against the (t', ε', Q_h)-UF-KOA-security of ID. We now build an adversary $\mathcal{B}$ against the $(t, \varepsilon, Q_{\text{CH}})$-PIMP-KOA security of ID, with $(t, \varepsilon, Q_{\text{CH}})$ as claimed.

Construction of $\mathcal{B}$. First, $\mathcal{B}$ obtains pk from its PIMP-KOA experiment which it forwards to $\mathcal{A}$. If $\mathcal{A}$ makes a query (R_i, m_i) to the random oracle, $\mathcal{B}$ makes a query $h_i \xleftarrow{\$} \text{CH}(R_i)$ and programs the random oracle $H(R_i, m_i) := h_i$. Eventually, $\mathcal{A}$ submits a forgery $(m, \sigma = (R, s))$, and terminates. We assume that $(R, m) \in \{(R_i, m_i)\}$, i.e., $H(R, m)$ was queries by $\mathcal{A}$. If not, $\mathcal{B}$ makes a dummy query to $H(R, m)$ which is simulated as described above. Hence, in total, there are $Q_{\text{CH}} := Q_h + 1$ queries to H. Let $i \in [Q_h + 1]$ be the unique index such that $(R_i, m_i) = (R, m)$. Adversary $\mathcal{B}$ outputs (i, s_i) and terminates. Note that (R_i, h_i, s_i) is a valid transcript and hence breaks PIMP-KOA security iff $\mathcal{A}$'s forgery is valid, establishing $\varepsilon = \varepsilon'$. The running time of $\mathcal{B}$ is roughly that of $\mathcal{A}$, hence $t' \approx t$. ■

The following lemma is a special case of Lemma 3.9 (with a slightly improved bound).

Lemma 3.7 (UF-KOA $\xrightarrow{\textbf{PRO}}$ UF-CMA). *Suppose* ID *is* HVZK *and has α bit min-entropy. If* SIG[ID] *is* (t, ε, Q_h)-UF-KOA *secure, then* SIG[ID] *is* $(t', \varepsilon', Q_s, Q_h)$-UF-CMA *secure in the programmable random oracle model, where*

$$\varepsilon' \leq \varepsilon + \frac{Q_h Q_s}{2^\alpha}, \quad t' \approx t,$$

and Q_s, Q_h are upper bounds on the number of signing and hash queries in the $\mathsf{UF\text{-}CMA}$ *experiment, respectively.*

Lemma 3.8 ($\mathsf{UF\text{-}KOA} \xrightarrow{\textbf{RSR}} \mathsf{MU\text{-}UF\text{-}KOA}$)**.** *Suppose* ID *is* RSR*. If* $\mathsf{SIG}[\mathsf{ID}]$ *is* (t, ε)-$\mathsf{UF\text{-}KOA}$ *secure, then* $\mathsf{SIG}[\mathsf{ID}]$ *is* (t', ε', N)-$\mathsf{MU\text{-}UF\text{-}KOA}$ *secure, where*

$$\varepsilon' = \varepsilon, \qquad t' \approx t.$$

Note that without the RSR property one can use the generic bounds from [GMS02] to obtain a non-tight bound with a loss of N.

Proof. Let $\mathcal{A}$ be an algorithm that breaks (t', ε', N)-$\mathsf{MU\text{-}UF\text{-}KOA}$ security of $\mathsf{SIG}[\mathsf{ID}]$. We will describe an adversary $\mathcal{B}$ invoking $\mathcal{A}$ that breaks (t, ε)-$\mathsf{UF\text{-}KOA}$ security of $\mathsf{SIG}[\mathsf{ID}]$ with (t, ε) as stated in the lemma. Adversary $\mathcal{B}$ is executed in the $\mathsf{UF\text{-}KOA}$ experiment and obtains a public-key pk.

Simulation of public-keys input to $\mathcal{A}$. For each $i \in [N]$, $\mathcal{B}$ generates $(pk_i, \mathbf{a}_i) \xleftarrow{\$} \mathsf{Rerand}(pk)$ by using the RSR property of ID. Then $\mathcal{B}$ runs $\mathcal{A}$ on input $(pk_1, \dots, pk_N)$.

Forgery. Eventually, $\mathcal{A}$ will submit its forgery $(i^*, m^*, \sigma^* := (R^*, s^*))$ in the $\mathsf{MU\text{-}UF\text{-}KOA}$ experiment. $\mathcal{B}$ computes $h^* = H(m^*, R^*)$ and runs $s \xleftarrow{\$} \mathsf{Tran}(pk, pk_{i^*}, \mathbf{a}_{i^*}, (R^*, h^*, s^*))$. By the RSR property of ID, the random variables (pk, R^*, h^*, s) and $(pk_{i^*}, R^*, h^*, s^*)$ are identically distributed. If σ^* is a valid signature on message m^* under pk_{i^*}, then (R^*, s) is also a valid signature on m^* under pk. Thus, we have $\varepsilon = \varepsilon'$. The running time t of $\mathcal{B}$ is t' plus the N times the time to run the Rerand and Tran algorithms of RSR. We again write $t' \approx t'$. ■

Lemma 3.9 ($\mathsf{MU\text{-}UF\text{-}KOA} \xrightarrow{\textbf{PRO}} \mathsf{MU\text{-}UF\text{-}CMA}$)**.** *Suppose* ID *is* HVZK *and has α bit min-entropy. If* $\mathsf{SIG}[\mathsf{ID}]$ *is* (t, ε, N, Q_h)-$\mathsf{MU\text{-}UF\text{-}KOA}$ *secure, then* $\mathsf{SIG}[\mathsf{ID}]$ *is* $(t', \varepsilon', N, Q_s, Q_h)$-$\mathsf{MU\text{-}UF\text{-}CMA}$ *secure in the programmable random oracle model, where*

$$\varepsilon' \leq 4\varepsilon + \frac{Q_h Q_s}{2^\alpha}, \quad t' \approx t,$$

and N is the number of users and Q_s and Q_h are upper bounds on the number of signing and hash queries in the $\mathsf{MU\text{-}UF\text{-}CMA}$ *experiment, respectively.*

Proof. Let $\mathcal{A}$ be an algorithm that breaks $(t', \varepsilon', N, Q_s, Q_h)$-$\mathsf{MU\text{-}UF\text{-}CMA}$ security of $\mathsf{SIG}[\mathsf{ID}]$. We will describe an adversary $\mathcal{B}$ invoking $\mathcal{A}$ that breaks (t, ε, N, Q_h)-$\mathsf{MU\text{-}UF\text{-}KOA}$ security of $\mathsf{SIG}[\mathsf{ID}]$ with (t, ε) as stated in the lemma. Adversary $\mathcal{B}$ is executed in the $\mathsf{MU\text{-}UF\text{-}KOA}$ experiment and obtains public-keys $(pk_1, \dots, pk_N)$, and has access to a random oracle H.

Preparation of public-keys. For each $i \in [N]$, adversary $\mathcal{B}$ picks a secret bit $b_i \xleftarrow{\$} \{0, 1\}$. If $b_i = 1$ then $\mathcal{B}$ defines $pk'_i := pk_i$, else $\mathcal{B}$ generates the key-pair $(pk'_i, sk'_i) \xleftarrow{\$} \mathsf{Gen}(\mathsf{par})$ itself. We note that all simulated public-keys are correctly distributed.

Adversary $\mathcal{B}$ runs $\mathcal{A}$ on input $(pk'_1, \dots, pk'_N)$ answering hash queries to random oracle H' and signing queries as follows.

Simulation of hash queries. A hash query $H'(R, m)$ is answered by $\mathcal{B}$ by querying its own hash oracle $H(R, m)$ and returning its answer.

Simulation of signing queries. On $\mathcal{A}$'s j-th signature query (i_j, m_j), $\mathcal{B}$ returns a signature σ_j on message m_j under pk_{i_j} according to the following case distinction.

- Case A: $b_{i_j} = 0$. In that case sk'_{i_j} is known to $\mathcal{B}$ and the signature is computed as $\sigma_j := (R_j, s_j) \xleftarrow{\$} \mathsf{Sign}(sk'_{i_j}, m_j)$. Note that this involves $\mathcal{B}$ making a hash query and defining $H'(R_j, m_j) := H(R_j, m_j)$.
- Case B: $b_{i_j} = 1$. In that case sk'_{i_j} is unknown to $\mathcal{B}$ and the signature is computed using the HVZK property of ID. Concretely, $\mathcal{B}$ runs $(R_j, h_j, s_j) \xleftarrow{\$} \mathsf{Sim}(pk'_{i_j})$. If hash value $H'(R_j, m_j)$ was already defined (via one of $\mathcal{A}$'s hash/signing queries) and $H'(R_j, m_j) \neq h_j$, $\mathcal{B}$ aborts. Otherwise, it defines the random oracle

$$H'(R_j, m_j) := h_j \tag{5}$$

  and returns $\sigma_j := (R_j, s_j)$, which is a correctly distributed valid signatures on m_j under pk_{i_j}. Note that by (5), $\mathcal{B}$ makes H and H' inconsistent, i.e., we have $H(R_j, m_j) \neq H'(R_j, m_j)$ with high probability. Also note that for each signing query, $\mathcal{B}$ aborts with probability at most $Q_h/2^\alpha$ because R_j has min-entropy α. Since the number of signing queries is bounded by Q_s, $\mathcal{B}$ aborts overall with probability at most $Q_h Q_s/2^\alpha$.

Forgery. Eventually, $\mathcal{A}$ will submit its forgery $(i^*, m^*, \sigma^* := (R^*, s^*))$. We assume that it is a valid forgery in the MU-UF-CMA experiment, i.e., for $h^* = H'(m^*, R^*)$ we have $\mathsf{V}(pk'_{i^*}, R^*, h^*, s^*) = 1$. Furthermore, it satisfies the freshness condition, i.e.,

$$(i^*, m^*) \notin \{(i_j, m_j) : j \in [Q_s]\}. \tag{6}$$

After receiving $\mathcal{A}$'s forgery, $\mathcal{B}$ computes a forgery for the MU-UF-KOA experiment according to the following case distinction.

- Case 1: There exists a $j \in [Q_s]$ such that $(m^*, R^*) = (m_j, R_j)$. (If there is more than one j, fix any of them.) In that case we have and $h^* = h_j$ and furthermore $i^* \neq i_j$ by the freshness condition (6).
  - Case 1a: $(b_{i^*} = 1)$ and $(b_{i_j} = 0)$. Then the hash value $h^* = H'(R^*, m^*)$ was not programmed by $\mathcal{B}$ in (5). That means $h^* = H'(R^*, m^*) = H(R^*, m^*)$ and $\mathcal{B}$ returns $(i^*, m^*, (R^*, s^*))$ as a valid forgery to its MU-UF-KOA experiment.
  - Case 1b: $(b_{i^*} = b_{i_j})$ or $(b_{i^*} = 0 \wedge b_{i_j} = 1)$. Then $\mathcal{B}$ aborts.

  Note that in case 1 we always have $i^* \neq i_j$ and therefore $\mathcal{B}$ does not abort with probability $1/4$ in which case it outputs a valid forgery.
- Case 2: For all $j \in [Q_s]$ we have: $(m^*, R^*) \neq (m_j, R_j)$.
  - Case 2a: $b_{i^*} = 1$. Then the hash value $h^* = H'(R^*, m^*)$ was not programmed by $\mathcal{B}$ in (5). That means $h^* = H'(R^*, m^*) = H(R^*, m^*)$ and $\mathcal{B}$ returns $(i^*, m^*, (R^*, s^*))$ as a valid forgery to its MU-UF-KOA experiment.
  - Case 2b: $b_{i^*} = 0$. Then $\mathcal{B}$ aborts.

  Note that in case 2, $\mathcal{B}$ does not abort with probability $1/2$ in which case it outputs a valid forgery.

Overall, $\mathcal{B}$ returns a valid forgery of MU-UF-KOA experiment with probability

$$\varepsilon \geq \min\left\{\frac{1}{4}, \frac{1}{2}\right\} \cdot \left(\varepsilon' - \frac{Q_h Q_s}{2^\alpha}\right) = \frac{1}{4}\left(\varepsilon' - \frac{Q_h Q_s}{2^\alpha}\right).$$

The running time of $\mathcal{B}$ is that of $\mathcal{A}$ plus the Q_s executions of Sim. We write $t' \approx t$. This completes the proof. ■

If s in ID is uniquely defined by (pk, R, h) (e.g., as in the Schnorr identification scheme), then one can show the above proof even implies MU-SUF-CMA security of SIG[ID]. The simulation of hash and signing queries is the same as in the above proof. Let (i^*, m^*, R^*, s^*) be $\mathcal{A}$'s forgery. The freshness condition of the MU-SUF-CMA experiment says that $(i^*, m^*, R^*, s^*) \notin \{(i_j, m_j, R_j, s_j) : j \in [Q_s]\}$. Together with the uniqueness of ID, this implies $(i^*, m^*, R^*) \notin \{(i_j, m_j, R_j) : j \in [Q_s]\}$. If $(i^*, m^*) \notin \{(i_j, m_j) : j \in [Q_s]\}$, then $\mathcal{B}$ can break MU-UF-KOA security by the same case distinction as in the proof above. Otherwise, we have $R^* \notin \{R_j : j \in [Q_s]\}$, in which case we can argue as in case 2.

4 Impossibility Results

In this section, we show that Theorems 3.1 and 3.2 from the previous section are optimal in the sense that the security reduction requires: rewinding (Lemma 4.1), security loss of at least $O(Q)$ (Lemma 4.3) and programmability of random oracles (Lemmas 4.5 and 4.6).

Let X and Y be some hard cryptographic problems, defined through a (possibly) interactive experiment. A black-box reduction $\mathcal{R}$ from X to Y is an algorithm that, given black-box access to an adversary $\mathcal{A}$ breaking problem Y, breaks problem X. If X and Y are security notions for identification or signatures schemes, then a reduction $\mathcal{R}$ is called key-preserving, if $\mathcal{R}$ only makes calls to $\mathcal{A}$ with the same pk that it obtained by its own problem X. All our reductions considered in this section are key-preserving.

Lemma 4.1 (KR-KOA $\overset{\textbf{non-rewind.}}{\not\longrightarrow}$ IMP-KOA)**.** *If there is a public-key preserving reduction $\mathcal{R}$ that $(t_\mathcal{R}, \varepsilon_\mathcal{R})$-breaks* KR-KOA *security of* ID *with* one-time black-box *access to an adversary $\mathcal{A}$ that $(t_\mathcal{A}, \varepsilon_\mathcal{A})$-breaks* IMP-KOA *security of* ID*, then there exists an algorithm $\mathcal{M}$ that $(t_\mathcal{M}, \varepsilon_\mathcal{M}, Q_\mathrm{O})$-breaks* IMP-AA *security of* ID*, where*

$$\varepsilon_{\mathcal{M}} \geq \varepsilon_{\mathcal{R}} - \frac{1}{|\mathsf{ChSet}|},\ t_{\mathcal{M}} \approx t_{\mathcal{R}}, Q_{\mathrm{O}} = 1.$$

Proof. Assuming the existence of a public-key preserving reduction $\mathcal{R}$ as above, we construct a meta-reduction $\mathcal{M}$ to break IMP-AA security of ID. $\mathcal{M}$ gets the public key pk of the IMP-AA challenge as input and has oracle access to $\mathrm{O} = \textsc{Prover}$, black-box accesses to $\mathcal{R}$ and simulates the adversary $\mathcal{A}$.

Construction of $\mathcal{M}(pk)$. $\mathcal{M}$ runs $\mathcal{R}(pk)$ and, upon receiving pk from $\mathcal{R}$, $\mathcal{M}$ simulates $\mathcal{A}(pk)$ as follows. First, $\mathcal{M}$ queries $R \xleftarrow{\$} \textsc{Prover}_1()$ provided by the IMP-AA experiment and returns R to $\mathcal{R}$. Upon receiving h from $\mathcal{R}$, $\mathcal{M}$ queries $s \xleftarrow{\$} \textsc{Prover}_2(1, h)$ provided by the IMP-AA experiment and returns s to $\mathcal{R}$ with probability $\varepsilon_{\mathcal{A}}$. After receiving sk from $\mathcal{R}$, $\mathcal{M}$ uses sk to impersonate a prover. First, $\mathcal{M}$ computes $(R^*, St) \xleftarrow{\$} \mathsf{P}(sk)$ and queries oracle $h^* \xleftarrow{\$} \textsc{Ch}(R^*)$ provided by the IMP-AA experiment. Next, $\mathcal{M}$ outputs $s^* \xleftarrow{\$} \mathsf{P}(sk, R^*, h^*, St)$ and terminates.

By the correctness of ID, (R^*, h^*, s^*) is a valid transcript and $(R^*, h^*, s^*) \neq (R, h, s)$ with probability at least $1 - 1/|\mathsf{ChSet}|$. We note that $\mathcal{M}$ perfectly simulates an $(t_{\mathcal{A}}, \varepsilon_{\mathcal{A}})$ adversary against IMP-KOA security. Thus, we have $\varepsilon_{\mathcal{M}} \geq \varepsilon_{\mathcal{R}} - 1/|\mathsf{ChSet}|$. ■

For our next impossibility result, we will require the following definition for identification schemes.

Definition 4.2 (Concurrent (Weak) Impersonation against Man-in-the-Middle Attacks). *A canonical identification* ID *is said to be* $(t, \varepsilon, Q_{\mathrm{CH}}, Q_{\mathrm{O}})$-IMP-MIM *secure (impersonation against man-in-the-middle attacks) if for all adversaries $\mathcal{A}$ running in time at most t and adaptively making at most Q_{O} queries to the prover oracle* Prover *and Q_{CH} queries to the challenge oracle* Ch*,*

$$\Pr\left[\begin{array}{l} \mathsf{V}(pk, R_{i^*}, h_{i^*}, s_{i^*}) = 1 \wedge (i^* \in [Q_{\mathrm{CH}}]) \\ \wedge (R_{i^*}, h_{i^*}, s_{i^*}) \notin \{(R'_j, h'_j, s'_j) \mid j \in [Q_{\mathrm{O}}]\} \end{array} \middle| \begin{array}{l} (pk, sk) \xleftarrow{\$} \mathsf{IGen}(\mathsf{par}) \\ (i^*, s_{i^*}) \xleftarrow{\$} \mathcal{A}^{\textsc{Prover}(\cdot),\textsc{Ch}(\cdot)}(pk) \end{array}\right] \leq \varepsilon,$$

where oracles Prover *and* Ch *are defined as in Definition 2.5. We define weak impersonation against man-in-the-middle attack (*wIMP-MIM*) by restricting $R_{i^*} \in \{R'_1, \ldots, R'_{Q_{\mathrm{O}}}\}$.*

The following generalizes a result by Seurin [Seu12] to canonical identification schemes.

Lemma 4.3 (IMP-KOA $\overset{\textbf{loss} <\mathbf{Q}}{\not\longrightarrow}$ PIMP-KOA)**.** *Suppose that* ID *has α bit min-entropy and there is a public-key preserving reduction $\mathcal{R}$ that $(t_{\mathcal{R}}, \varepsilon_{\mathcal{R}})$-breaks* IMP-KOA *security of* ID *with n-*time black-box *access to an adversary $\mathcal{A}$ that $(t_{\mathcal{A}}, \varepsilon_{\mathcal{A}}, Q_{\mathrm{CH}})$-breaks* PIMP-KOA *security of* ID*. Then there exists an algorithm $\mathcal{M}$ that $(t_{\mathcal{M}}, \varepsilon_{\mathcal{M}}, 1, Q_{\mathrm{O}} = nQ_{\mathrm{CH}})$-breaks* IMP-MIM *security of* ID*, where*

$$\varepsilon_{\mathcal{M}} \geq \varepsilon_{\mathcal{R}} - \frac{n \ln\left((1 - \varepsilon_{\mathcal{A}})^{-1}\right)}{Q_{\mathrm{CH}}} - \frac{n}{|\mathsf{ChSet}|} - \frac{n}{2^{\alpha}},\ t_{\mathcal{M}} \approx t_{\mathcal{R}}.$$

We note that the Schnorr identification scheme is wIMP-MIM but not IMP-MIM-secure (cf. Section 5.1).

Proof. Assuming the existence of a public-key preserving reduction $\mathcal{R}$, we construct a meta-reduction $\mathcal{M}$ to break IMP-MIM security of ID (see Figure 3). $\mathcal{M}$ inputs public key pk of the IMP-MIM challenger, has black-box accesses to $\mathcal{R}$ and simulates the adversary $\mathcal{A}$ while interacting within $Q_{\mathrm{O}} = nQ_{\mathrm{CH}}$ many MIM rounds.

W.l.o.g. we can assume that our adversary $\mathcal{A}$ never accesses its random coins. Instead, it generates pseudorandomness directly using a PRF, where the key k of PRF is part of the description of $\mathcal{A}$. Adversary $\mathcal{A}$'s randomness is derived from its current view using the PRF. As we assume that $\mathcal{R}$ has only black box access to $\mathcal{A}$, it can not access key k and hence it can not distinguish $\mathcal{A}$'s pseudorandom randomness from uniform randomness by observing the outputs of $\mathcal{A}$.

Construction of $\mathcal{M}(pk)$. $\mathcal{M}$ runs $\mathcal{R}(pk)$ who is interacting with a simulated $\mathcal{A}(pk)$. (Recall that $\mathcal{R}$ is public-key preserving, so it always executed $\mathcal{A}$ on pk.) $\mathcal{R}$ can execute $\mathcal{A}$ at most n times and hence rewinds it at most $n - 1$ times to any desired state. In the simulation of $\mathcal{A}$ described below we make the explicit convention that $\mathcal{M}$ always keeps the simulation of $\mathcal{A}$ consistent with previous executions. That is, as long as there exists a $j' < j$ such that for all $i' < i$, $h_{j,i'} = h_{j',i'}$, then $\mathcal{M}$ will also use $R_{j,i} = R_{j',i}$ and $c_{j,i} = c_{j',i}$.

Upon receiving pk from $\mathcal{R}$, $\mathcal{M}$ simulates the j-th execution or rewind ($j \in [n]$) of $\mathcal{A}(pk)$ as follows.

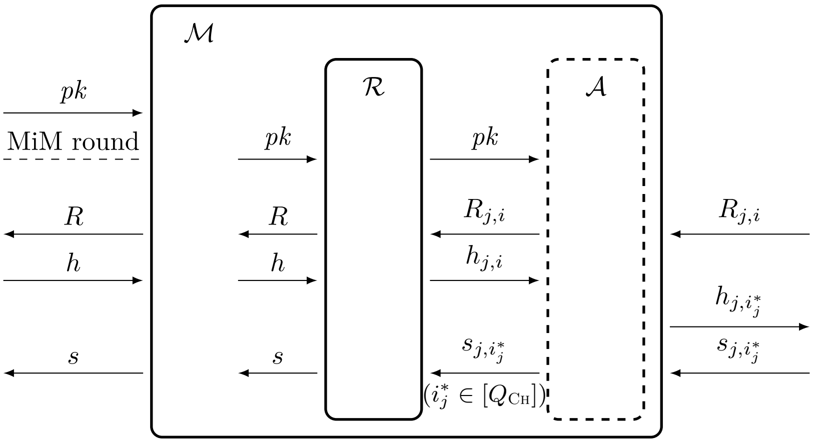


Figure 3: Meta-reduction $\mathcal{M}$ uses $\mathcal{R}$ to break the $\mathsf{IMP\text{-}MIM}$ security in n MIM rounds. n is the total amount of executions of $\mathcal{A}$ performed by $\mathcal{R}$. For every MIM round $j \in [n]$, $\mathcal{A}$ picks an $i_j^* \in [Q_{\text{CH}}]$ and forwards a valid response s_{j,i_j^*}. $\mathcal{M}$ fails when $\mathcal{R}$ fails, $(R, h, s) = (R_{i_j^*}, h_{i_j^*}, s_{i_j^*})$ for some $j \in [n]$. It also fails with some probability when $\mathcal{A}$ gets rewinded on a different h_{j,i_j^*} after having requested s_{j,i_j^*}.

- First, $\mathcal{M}$ sets a flag $b_j := 0$. The flag b_j will be switched of 1 once $\mathcal{M}$ has obtained one valid transcript from the Prover oracle.
- To simulate the i-th query to the challenge oracle ($i \in [Q_{\text{CH}}]$), $\mathcal{M}$ starts an interaction with a new prover: $\mathcal{M}$ calls $R_{j,i} \xleftarrow{\$} \text{Prover}_1()$ and forwards it to $\mathcal{R}$, which will reply with an arbitrary $h_{j,i} \in \mathsf{ChSet}$. If $b_j = 1$, $\mathcal{M}$ sets $c_{j,i} := 0$. Otherwise, $\mathcal{M}$ flips a biased coin $c_{j,i}$ with $\Pr[c_{j,i} = 1] = \mu$, where μ will be defined later.
  Case 1: $c_{j,i} = 1$. If there is an index $j' < j$ with $R_{j',i} = R_{j,i}$, $h_{j',i} \neq h_{j,i}$, and $c_{j',i} = 1$, then $\mathcal{M}$ aborts its attempt to break $\mathsf{IMP\text{-}MIM}$ security of ID. Otherwise, it defines $i_j^* := i$ and requests $s_{j,i_j^*} \xleftarrow{\$} \text{Prover}_2(I_{j,i_j^*}, h_{j,i_j^*})$, where $I_{j,i_j^*} = (j-1) \cdot Q_{\text{CH}} + i_{j^*}$ refers to the I_{j,i_j^*}'s query to the Prover$_1$ oracle from which $\mathcal{M}$ obtained challenge h_{j,i_j^*}. Note that $\mathcal{M}$ now obtained one transcript $(R_{j,i_j^*}, h_{j,i_j^*}, s_{j,i_j^*})$ from the Prover oracle and therefore sets $b_j := 1$.
  Case 2: $c_{j,i} = 0$. $\mathcal{M}$ does nothing.
- After Q_{CH} simulated challenge queries, $\mathcal{M}$ sets $(i_j^*, s_{j,i_j^*}) := (\bot, \bot)$ if i_j^* is undefined. Finally, $\mathcal{M}$ returns (i_j^*, s_{j,i_j^*}) to $\mathcal{R}$.

This completes the simulation of the j-th execution of $\mathcal{A}$.

At some point $\mathcal{R}$ makes a query $\text{Ch}(R)$, which $\mathcal{M}$ forwards to its own Ch, receiving h. Finally, $\mathcal{R}$ outputs s and terminates. $\mathcal{M}$ also outputs s and terminates. This completes the description of $\mathcal{M}$.

Analysis of $\mathcal{M}$.

We define Bad_1 as the event that the transcript (R, h, s) output by $\mathcal{R}$ does not satisfy the freshness condition $(R, h, s) \notin \{(R_{j,i}, h_{j,i}, s_{j,i}) \mid (j, i) \in [n] \times [Q_{\text{CH}}]\}$ of the $\mathsf{IMP\text{-}MIM}$ security experiment. Note that $s_{j,i} \neq \bot$ only if $i = i_j^*$ and therefore to consider the case when $i = i_j^*$.

$$\begin{aligned}\Pr[\text{Bad}_1] &= \Pr[\exists j \in [n] : (R, h, s) = (R_{j,i_j^*}, h_{j,i_j^*}, s_{j,i_j^*})] \\ &\leq \Pr[\exists j \in [n] : (R, h) = (R_{j,i_j^*}, h_{j,i_j^*})].\end{aligned}$$

We let $(j_0, i_0) \in [n] \times [Q_{\text{CH}}]$ be the unique index such that $\mathcal{R}$ makes its single query $\text{Ch}(R)$ after receiving R_{j_0,i_0} but before receiving R_{j_0,i_0+1}.

$$\begin{aligned}\Pr[\exists j \in [n] : (R, h) = (R_{j,i_j^*}, h_{j,i_j^*})] \leq\ & \Pr[\exists (j, i_j^*) \neq (j_0, i_0) : (R, h) = (R_{j,i_j^*}, h_{j,i_j^*})] && (7) \\ & + \Pr[(j, i_j^*) = (j_0, i_0)]. && (8)\end{aligned}$$

We bound the probabilities (7) and (8) individually. To bound (8), only a single query is considered.

Therefore

$$\Pr[(j, i_j^*) = (j_0, i_0)] \quad = \quad \Pr[c_{j_0,i_0} = 1] \le \mu.$$

To bound (7), we define a natural order on the set $[n] \times [Q_{\text{CH}}]$ via $(j, i) < (j_0, i_0)$ iff $R_{j,i}$ was received before R_{j_0,i_0}, i.e, $(j-1)Q_{\text{CH}} + i < (j_0 - 1)Q_{\text{CH}} + i_0$. Note that $\mathcal{R}$ chooses h_{j,i_j^*} for $(j, i_j^*) < (j_0, i_0)$ before seeing $h \overset{\$}{\leftarrow} \mathsf{ChSet}$. Furthermore, R is fixed for $(j, i_j^*) > (j_0, i_0)$ while $R_{j,i_j^*} \overset{\$}{\leftarrow} \textsc{Prover}_1()$ has at least α bits of min-entropy. Therefore by splitting the probabilities and using a union bound

$$\begin{aligned}
&\Pr[\exists (j, i_j^*) \neq (j_0, i_0) : (R, h) = (R_{j,i_j^*}, h_{j,i_j^*})] \\
&\le \quad \Pr[\exists (j, i_j^*) < (j_0, i_0) : (R, h) = (R_{j,i_j^*}, h_{j,i_j^*})] + \Pr[\exists (j, i_j^*) > (j_0, i_0) : (R, h) = (R_{j,i_j^*}, h_{j,i_j^*})] \\
&\le \quad \Pr[\exists (j, i_j^*) < (j_0, i_0) : h = h_{j,i_j^*}] + \Pr[\exists (j, i_j^*) > (j_0, i_0) : R = R_{j,i_j^*}] \\
&\le \quad \frac{j_0}{|\mathsf{ChSet}|} + \frac{n - j_0 + 1}{2^\alpha} \le \frac{n}{|\mathsf{ChSet}|} + \frac{n}{2^\alpha}.
\end{aligned}$$

Overall, this yields

$$\Pr[\text{Bad}_1] \le \frac{n}{|\mathsf{ChSet}|} + \mu + \frac{n}{2^\alpha}.$$

Next, we define Bad_2 as the event that $\mathcal{M}$ aborts. By a union bound we get

$$\begin{aligned}
\Pr[\text{Bad}_2] \quad &= \quad \Pr[\exists j \in [n], j' < j, i \in [Q_{\text{CH}}] : R_{j',i} = R_{j,i} \land h_{j',i} \neq h_{j,i} \land c_{j,i} = c_{j',i} = 1] \\
&= \quad \Pr[\exists j \in [n], j' < j : R_{j',i_j^*} = R_{j,i_j^*} \land h_{j',i_j^*} \neq h_{j,i_j^*} \land c_{j',i_j^*} = 1] \\
&\le \quad \Pr[\exists j \in [n], j' < j : c_{j',i_j^*} = 1] \le (n-1)\mu.
\end{aligned}$$

Chioce of μ. We now choose μ such that on one side $\mathcal{A}$ forges with probability $\varepsilon_{\mathcal{A}}$ and on the other side the probability that Bad_1 or Bad_2 happen is bounded. We set

$$\mu = 1 - (1 - \varepsilon_{\mathcal{A}})^{1/Q_{\text{CH}}}$$

for a desired success probability $0 < \varepsilon_{\mathcal{A}} < 1$ of $\mathcal{A}$ and Q_{CH} queries. Note that for an execution $j \in [n]$ that unless for all $i \in [Q_{\text{CH}}]$ we have $c_{j,i} = 0$, $\mathcal{A}$ will always send a valid transcript and break the $\mathsf{PIMP\text{-}KOA}$ security. Let $\overline{\mu} := (1 - \mu)$. For any execution $j \in [n]$, $\mathcal{A}$ has success probability

$$\Pr[\exists i \in [Q_{\text{CH}}] : c_{j,i} = 1] = \sum_{k=1}^{Q_{\text{CH}}} \mu(1-\mu)^{k-1} = \sum_{k=1}^{Q_{\text{CH}}} (\overline{\mu}^{k-1} - \overline{\mu}^k) = 1 - (1-\mu)^{Q_{\text{CH}}} = \varepsilon_{\mathcal{A}}.$$

Finally, we can bound the success probability of $\mathcal{M}$

$$\Pr[\text{Bad}_1 \land \text{Bad}_2] \le n \cdot \mu + \frac{n}{|\mathsf{ChSet}|} + \frac{n}{2^\alpha} \le \frac{n \ln\left((1 - \varepsilon_{\mathcal{A}})^{-1}\right)}{Q_{\text{CH}}} + \frac{n}{|\mathsf{ChSet}|} + \frac{n}{2^\alpha},$$

where the bound $\mu \le \ln((1 - \varepsilon_{\mathcal{A}})^{-1})/Q_{\text{CH}}$ was proved in [Seu12, Lemma 1]. Therefore we have

$$\varepsilon_{\mathcal{M}} \ge \varepsilon_{\mathcal{R}} - \frac{n \ln\left((1 - \varepsilon_{\mathcal{A}})^{-1}\right)}{Q_{\text{CH}}} - \frac{n}{|\mathsf{ChSet}|} - \frac{n}{2^\alpha},\ t_{\mathcal{M}} \approx t_{\mathcal{R}} \approx n t_{\mathcal{A}}$$

which concludes the proof of the lemma. ■

For a precise analysis of the function $\ln\left((1 - \varepsilon_{\mathcal{A}})^{-1}\right)$, we refer to [Seu12]. For our purpose, it is sufficient that for a concrete choice of $\varepsilon_{\mathcal{A}}$, there is a constant c such that $c \cdot \varepsilon_{\mathcal{A}} = \ln\left((1 - \varepsilon_{\mathcal{A}})^{-1}\right)$. Hence Lemma 4.3 gives roughly $\varepsilon_{\mathcal{M}} \ge \varepsilon_{\mathcal{R}} - c \cdot n/Q_{\text{CH}} \cdot \varepsilon_{\mathcal{A}}$ for a suitable choice of $\varepsilon_{\mathcal{A}}$. Therefore $\varepsilon_{\mathcal{R}}$ can be at most $c \cdot n/Q_{\text{CH}} \cdot \varepsilon_{\mathcal{A}}$. Otherwise $\mathcal{M}$ would break $\mathsf{IMP\text{-}MIM}$ security of ID with $\varepsilon_{\mathcal{M}} > 0$.

It is easy to see that the meta-reduction of the proof of Lemma 4.3 just forwards all $R_{j,i}$ received during the Man-in-the-Middle attack and R send by $\mathcal{R}$. So if $\mathcal{R}$ is furthermore randomness-preserving, i.e., it chooses $R \in \{R_{1,1}, \ldots, R_{n,Q_{\text{CH}}}\}$, then $\mathcal{M}$ attacks $\mathsf{wIMP\text{-}MIM}$-security of ID. This observation is formalized in the following corollary.

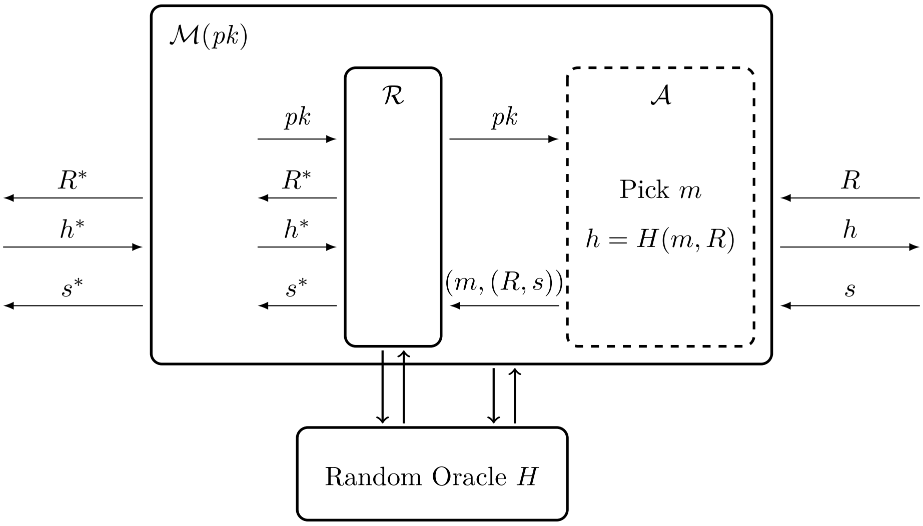


Figure 4: Meta-reduction $\mathcal{M}$ runs $\mathcal{R}$ to break IMP-AA security in the non-programmable random oracle model, where both $\mathcal{M}$ and $\mathcal{R}$ have oracle access to the same external random oracle H. $\mathcal{M}$ simulates an adversary $\mathcal{A}$ that $(t_{\mathcal{A}}, \varepsilon_{\mathcal{A}}, Q_h)$-breaks UF-KOA security of SIG[ID] (which is in the dashed box) and answers the oracle queries of $\mathcal{R}$.

Corollary 4.4. *If* ID *has α bit min-entropy and there exists a public key and randomness preserving reduction $\mathcal{R}$ that $(t_{\mathcal{R}}, \varepsilon_{\mathcal{R}})$-breaks* IMP-KOA *security of* ID *with n-*time black-box *access to an adversary $\mathcal{A}$ that $(t_{\mathcal{A}}, \varepsilon_{\mathcal{A}}, Q_{\text{CH}})$-breaks* PIMP-KOA *security of* ID*, then there exists an algorithm $\mathcal{M}$ that $(t_{\mathcal{M}}, \varepsilon_{\mathcal{M}}, 1, Q_{\text{O}} = nQ_{\text{CH}})$-breaks* wIMP-MIM *security of* ID*, where*

$$\varepsilon_{\mathcal{M}} \geq \varepsilon_{\mathcal{R}} - \frac{n \ln\left((1-\varepsilon_{\mathcal{A}})^{-1}\right)}{Q_{\text{CH}}} - \frac{n}{|\mathsf{ChSet}|} - \frac{n}{2^{\alpha}},\ t_{\mathcal{M}} \approx t_{\mathcal{R}}.$$

Lemma 4.5 (IMP-KOA $\overset{\textbf{NPRO}}{\not\longrightarrow}$ UF-KOA)**.** *If there exists a public key preserving reduction $\mathcal{R}$ in the non-programmable random oracle (NPRO) model that $(t_{\mathcal{R}}, \varepsilon_{\mathcal{R}})$-breaks* IMP-KOA *security of* ID *with black-box access to an adversary $\mathcal{A}$ that $(t_{\mathcal{A}}, \varepsilon_{\mathcal{A}}, Q_h)$-breaks* UF-KOA *security of* SIG[ID]*, then there exists an algorithm $\mathcal{M}$ that $(t_{\mathcal{M}}, \varepsilon_{\mathcal{M}}, 1)$-breaks* IMP-AA*-security of* ID*, where*

$$\varepsilon_{\mathcal{M}} \geq \varepsilon_{\mathcal{R}} - \frac{1}{|\mathsf{ChSet}|},\ t_{\mathcal{M}} \approx t_{\mathcal{R}}.$$

Proof. Assuming the existence of a public key preserving reduction $\mathcal{R}$ as above, we construct a meta-reduction $\mathcal{M}$ to break IMP-AA security of ID. Figure 4 gives a pictorial overview of it $\mathcal{M}$. $\mathcal{M}$ obtains the public key pk from the IMP-AA experiment and has oracle access to Prover, black-box accesses to $\mathcal{R}$ and simulates the adversary $\mathcal{A}$. Additionally, both $\mathcal{M}$ and $\mathcal{R}$ get access to the same external random oracle H, in the NPRO model.

Construction of $\mathcal{M}(pk)$. $\mathcal{M}$ runs $\mathcal{R}(pk)$ and, upon receiving pk from $\mathcal{R}$, $\mathcal{M}$ simulates $\mathcal{A}(pk)$ as follows. First, $\mathcal{M}$ queries $R \xleftarrow{\$} \textsc{Prover}_1()$ to the IMP-AA experiment . Next, $\mathcal{M}$ picks an arbitrary message m, queries $h = H(m, R)$ to the random oracle, and $s \xleftarrow{\$} \textsc{Prover}(1, h)$ to the IMP-AA experiment. With probability $\varepsilon_{\mathcal{A}}$, $\mathcal{M}$ returns $(m, (R, s))$ as a forgery to $\mathcal{R}$.

Upon receiving a challenge query $\textsc{Ch}_{\mathcal{R}}(R^*)$ query from $\mathcal{R}$, $\mathcal{M}$ answers with $h^* \xleftarrow{\$} \textsc{Ch}_{\mathcal{M}}(R^*)$, where $\textsc{Ch}_{\mathcal{M}}$ is provided by the IMP-AA experiment. Finally, $\mathcal{R}$ outputs s^* to break IMP-KOA security and terminates. $\mathcal{M}$ also outputs s^* to its IMP-AA experiment and terminates. We note that $(R^*, h^*, s^*) = q(R, h, s)$ with probability $1/|\mathsf{ChSet}|$, since h^* is a random challenge chosen by the IMP-AA experiment and h is the response of a random oracle query. Thus, if s^* breaks IMP-KOA security, then s^* breaks

IMP-AA security. Moreover, $\mathcal{M}$ perfectly simulates an adversary that $(t_{\mathcal{A}}, \varepsilon_{\mathcal{A}})$-breaks UF-KOA security. This establishes $\varepsilon_{\mathcal{M}} \geq \varepsilon_{\mathcal{R}} - 1/|\mathsf{ChSet}|$. ■

By Lemma 3.5, Lemma 4.5 implies that there is no reduction from PIMP-KOA to UF-KOA in the non-programmable random oracle model.

The following simple lemma actually holds for any signature scheme SIG.

Lemma 4.6 (UF-KOA $\overset{\textbf{NPRO}}{\not\longrightarrow}$ UF-CMA)**.** *Suppose that there is a public-key preserving reduction $\mathcal{R}$ in the non-programmable random oracle (NPRO) model that $(t_{\mathcal{R}}, \varepsilon_{\mathcal{R}}, Q_s, Q_h)$-breaks* UF-KOA *security of* SIG *with black-box access to an adversary $\mathcal{A}$ that $(\varepsilon_{\mathcal{A}}, t_{\mathcal{A}}, Q_h)$-breaks* UF-CMA *security of* SIG*. Then there exists an algorithm $\mathcal{M}$ that $(\varepsilon_{\mathcal{M}}, t_{\mathcal{M}})$-breaks* UF-KOA *security of* SIG*, where*

$$\varepsilon_{\mathcal{M}} \geq \varepsilon_{\mathcal{R}},\ t_{\mathcal{M}} \approx t_{\mathcal{R}}.$$

Proof. Assuming the existence of a public key preserving reduction $\mathcal{R}$ as above, we construct a meta-reduction $\mathcal{M}$ to break UF-KOA security of SIG[ID]. $\mathcal{M}$ gets the public key pk from the UF-KOA experiment and simulates an adversary $\mathcal{A}$. Additionally, both $\mathcal{M}$ and $\mathcal{R}$ get access to the same external random oracle H, in the NPRO model.

Construction of $\mathcal{M}(pk)$. $\mathcal{M}$ runs $\mathcal{R}(pk)$ and, upon receiving pk from $\mathcal{R}$, $\mathcal{M}$ make a signing query on $m \xleftarrow{\$} \mathcal{M}$ to $\mathcal{R}$. Upon receiving the signature $\sigma = (R, s)$, $\mathcal{M}$ terminates and returns (m, σ) as a UF-KOA forgery. As both $\mathcal{M}$ and $\mathcal{R}$ have access to the same random oracle, (m, σ) is a valid forgery in UF-KOA experiment. Thus, we have $\varepsilon_{\mathcal{M}} \geq \varepsilon_{\mathcal{R}}$. ■

Remark 4.7. All the reductions considered in this section are key-preserving which is the main restriction of our results. If pk and R are elements from some multiplicative group $\mathbb{G}$ of prime order p, then we can extend our previous techniques to exclude the larger class of algebraic reductions. A reduction is algebraic, if for all group elements h output by the reduction, their respective representation is known. That is, if at some point of its execution the reduction holds group elements $g_1, \ldots, g_n \in \mathbb{G}$ and outputs a new group element h, then it also knows it representation meaning it also outputs $(\alpha_1, \ldots, \alpha_n) \in \mathbb{Z}_p^n$ satisfying $h = \prod g_i^{\alpha_i}$. Note that key-preserving and randomness-preserving reductions are a special case of algebraic reductions.

5 Example Instantiations

In this section we consider two examples of important identification schemes, namely the ones by Schnorr [Sch91], by Katz-Wang [KW03] and by Guillou-Quisquater [GQ90]. We use our framework to derive tight security bounds and concrete parameters for the corresponding Schnorr/Katz-Wang/Guillou-Quiquater signature schemes.

5.1 Schnorr Identification/Signature Scheme

5.1.1 Schnorr's Identification Scheme

The well-known Schnorr's identification scheme is one of the most important examples of our framework. For completeness we show that Schnorr's identification has large min-entropy, special soundness (SS), honest-verifier zero-knowledge (HVZK), random-self reducibility (RSR) and key-recovery security (KR-KOA) based on the discrete logarithm problem (DLOG). Moreover, based on the one-more discrete logarithm problem (OMDL), Schnorr's identification is actively secure (IMP-AA) [BP02] and weakly secure against man-in-the-middle attack (wIMP-MIM) (Lemma 5.5).

Let $\mathsf{par} := (p, g, \mathbb{G})$ be a set of system parameters, where $\mathbb{G} = \langle g \rangle$ is a cyclic group of prime order p with a hard discrete logarithm problem. Examples of groups $\mathbb{G}$ include appropriate subgroups of certain elliptic curve groups, or subgroups of $\mathbb{Z}_q^*$. The Schnorr identification scheme $\mathsf{ID_S} := (\mathsf{IGen}, \mathsf{P}, \mathsf{ChSet}, \mathsf{V})$ is defined as follows.

```
$\mathsf{IGen}(\mathsf{par})$:
    $sk := x \xleftarrow{\$} \mathbb{Z}_p$
    $pk := X = g^x$
    $\mathsf{ChSet} := \{0,1\}^n$
    Return $(pk, sk)$

$\mathsf{V}(pk, R, h, s)$:
    If $R = g^s \cdot X^{-h}$ then return 1
    Else return 0

$\mathsf{P}_1(sk)$:
    $r \xleftarrow{\$} \mathbb{Z}_p$; $R = g^r$
    $St := r$
    Return $(R, St)$

$\mathsf{P}_2(sk, R, h, St)$:
    Parse $St = r$
    Return $s = x \cdot h + r \bmod p$
```

We recall the DLOG and OMDL assumptions.

Definition 5.1 (Discrete Logarithm Assumption). *The discrete logarithm problem* DLOG *is* (t, ε)*-hard in* $\mathsf{par} = (p, g, \mathbb{G})$ *if for all adversaries* $\mathcal{A}$ *running in time at most* t*,*

$$\Pr\left[\ g^x = X \ \middle|\ X \xleftarrow{\$} \mathbb{G}; x \xleftarrow{\$} \mathcal{A}(X)\ \right] \leq \varepsilon.$$

Lemma 5.2. $\mathsf{ID_S}$ *is a canonical identification with* $\alpha = \log p$ *bit min-entropy and it is unique, has special soundness (*SS*), honest-verifier zero-knowledge (*HVZK*) and is random-self reducible (*RSR*). Moreover, if* DLOG *is* (t, ε)*-hard in* $\mathsf{par} = (p, g, \mathbb{G})$ *then* $\mathsf{ID_S}$ *is* (t, ε)*-*$\mathsf{KR\text{-}KOA}$ *secure.*

Proof. The correctness of $\mathsf{ID_S}$ is straightforward to verify. We note that $R \xleftarrow{\$} \mathsf{P}_1(sk)$ is uniformly random over $\mathbb{G}$. Hence, ID has $\log|\mathbb{G}| = \log p$ bit min-entropy. We show the other properties as follows.

Uniqueness. For all $(X, x) \in \mathsf{IGen}(\mathsf{par})$, $R := g^r \in \mathsf{P}_1(sk)$ and $h \in \{0,1\}^n$, the value $s \in \mathbb{Z}_p$ satisfying $g^s = X^h R \Leftrightarrow s = xh + r$ is uniquely defined.

Special Soundness (SS). Given two accepting transcripts (R, h, s) and (R, h', s') with $h \neq h'$, we define an extractor algorithm $\mathsf{Ext}(X, R, h, s, h', s') := x^* := (s - s')/(h - h')$ such that, for all $(X := g^x, x) \in \mathsf{IGen}(\mathsf{par})$, we have $\Pr[g^{x^*} = X] = 1$, since we have $R = g^s X^{-h} = g^{s'} X^{-h'}$ and then $X = g^{(s-s')/(h-h')}$.

Honest-verifier Zero-knowledge (HVZK). Given public key X, we let $\mathsf{Sim}(X)$ first sample $h \xleftarrow{\$} \{0,1\}^n$ and $s \xleftarrow{\$} \mathbb{Z}_p$ and then output $(R := g^s X^{-h}, h, s)$. Clearly, (R, h, s) is a real transcript, since s is uniformly random over $\mathbb{Z}_p$ and R is the unique value satisfying $R = g^s X^{-h}$.

Random-self Reducibility (RSR). Algorithm Rerand and two deterministic algorithm Derand and Tran are defined as follows:

- $\mathsf{Rerand}(X)$ chooses $\mathbf{a}' \xleftarrow{\$} \mathbb{Z}_p$ and outputs $(X' := X \cdot g^{\mathbf{a}'}, \mathbf{a}')$. We have that, for all $(X, x) \in \mathsf{IGen}(\mathsf{par})$, X' is uniform and has the same distribution as X'', where $(X'', x'') \xleftarrow{\$} \mathsf{IGen}(\mathsf{par})$.
- $\mathsf{Derand}(X, X', x', \mathbf{a}')$ outputs $x^* = x' - \mathbf{a}'$. We have, for all $(X', \mathbf{a}') \xleftarrow{\$} \mathsf{Rerand}(X := g^x)$ and $(X', x') \in \mathsf{IGen}(\mathsf{par})$, $X' = g^{x'}$ and $x' = x + \mathbf{a}'$ and thus $x^* = x$.
- $\mathsf{Tran}(X, X', \mathbf{a}', (R', h', s'))$ outputs $s = s' - \mathbf{a}' \cdot h'$. We have, for all $(X', \mathbf{a}') \in \mathsf{Rerand}(X := g^x)$, if (R', h', s') is valid with respect to $X' := g^{x + \mathbf{a}'}$ then $s = s' - \mathbf{a}' \cdot h' = (x + \mathbf{a}')h' + r - \mathbf{a}' \cdot h' = xh' + r$ and (R', h', s) is valid with respect to X.

Key-recovery against Key-only Attack ($\mathsf{KR\text{-}KOA}$). $\mathsf{KR\text{-}KOA}$-security for ID is exactly the DLOG assumption. ■

Definition 5.3 (One-more Discrete Logarithm Assumption [BNPS03]). *We says that* OMDL *is* (t, ε, Q)*-hard in* $\mathsf{par} = (p, g, \mathbb{G})$ *if for all adversaries* $\mathcal{A}$ *running in time at most* t *and adaptively making at most* Q *queries to the discrete logarithm oracle* DL*,*

$$\Pr\left[\ \textit{For } i \in [Q+1] : X_i = g^{x_i} \ \middle|\ \begin{array}{l} X_1, \ldots, X_{Q+1} \xleftarrow{\$} \mathbb{G} \\ (x_1, \ldots, x_{Q+1}) \xleftarrow{\$} \mathcal{A}^{\mathrm{DL}(\cdot)}(X_1, \ldots, X_{Q+1}) \end{array} \right] \leq \varepsilon,$$

where on input arbitrary group element Y *the discrete logarithm oracle* DL *returns* $y \in \mathbb{Z}_p$ *such that* $g^y = Y$*.*

Lemma 5.4 (Theorem 5.1 in [BP02]). *If the* OMDL *problem is* (t, ε, Q)*-hard then* $\mathsf{ID_S}$ *is* $(t', \varepsilon', Q_{\mathrm{O}})$*-*$\mathsf{IMP\text{-}AA}$ *secure, where* $\varepsilon' \leq \sqrt{\varepsilon} + 1/p$*,* $t \approx 2t'$*, and* $Q_{\mathrm{O}} = Q$*.*

We now show that the Schnorr identification scheme is weakly IMP-MIM secure based on one-more discrete logarithm assumption.

Lemma 5.5. *If* OMDL *problem is* (t, ε, Q)*-hard then* $\mathsf{ID_S}$ *is* $(t', \varepsilon', Q_{\mathrm{CH}}, Q_{\mathrm{O}})$-wIMP-MIM *secure, where*

$$\varepsilon' = \varepsilon, \quad t' \approx t, \quad Q_{\mathrm{O}} = Q.$$

Proof. Let $\mathcal{A}$ be an algorithm that breaks $(t', \varepsilon', Q_{\mathrm{CH}}, Q_{\mathrm{O}})$-wIMP-MIM security of $\mathsf{ID_S}$. We will describe an adversary $\mathcal{B}$ invoking $\mathcal{A}$ that (t, ε, Q)-breaks OMDL with (t, ε, Q) as stated in the theorem. Adversary $\mathcal{B}$ obtains $X_1, \ldots, X_{Q+1}$, and has access to a discrete logarithm oracle $\textsc{Dl}$. $\mathcal{B}$ runs $\mathcal{A}$ on input $pk := X := X_{Q+1}$ and answers the adaptive $\textsc{Prover}$ and $\textsc{Ch}$ queries as follows:

- On the j-th $\textsc{Prover}_1()$ query ($j \in [Q_{\mathrm{O}}]$), $\mathcal{B}$ returns $R'_j := X_j$.
- On the j-th $\textsc{Prover}_2(j, h'_j)$ query, $\mathcal{B}$ queries and returns $s'_j = \textsc{Dl}(X^{h'_j} \cdot R'_j)$.
- On the i-th $\textsc{Ch}(R_i)$ query, $\mathcal{B}$ chooses a random $h_i \xleftarrow{\$} \mathsf{ChSet}$ and returns h_i.

Note that weak IMP-MIM security requires that for all i, j, we have $R_i = R'_j$ for some R'_j previously returned by the $\textsc{Prover}_1()$ oracle.

Eventually, $\mathcal{A}$ returns (i^*, s_{i^*}) and terminates. We can assume that $\mathcal{A}$ has made the queries $\textsc{Prover}_2(j, h'_j)$ for all $j \in [Q_{\mathrm{O}}]$. If not $\mathcal{B}$ makes the dummy query $\textsc{Prover}_2(j, h'_j)$ for an arbitrary $h'_j \neq h_{i^*}$ to obtain a valid transcript (R'_j, h'_j, s'_j) for all $j \in [Q_{\mathrm{O}}]$. So in total, $\mathcal{B}$ made exactly Q_{O} calls to the $\textsc{Dl}$ oracle.

$\mathcal{A}$ wins if $(R_{i^*}, h_{i^*}, s_{i^*})$ is a valid transcript, $(R_{i^*}, h_{i^*}, s_{i^*}) \notin \{(R'_j, h'_j, s'_j) \mid j \in [Q_{\mathrm{O}}]\}$, and $R_{i^*} = R'_{j^*}$, for some index j^*. (If there exists more than one index j^*, we fix an arbitrary one.) From the above observations we conclude that $\mathcal{B}$ knows two valid transcripts, $(R_{i^*}, h_{i^*}, s_{i^*})$ and $(R'_{j^*} = R_{i^*}, h'_{j^*}, s'_{j^*})$ satisfying $(h_{i^*}, s_{i^*}) \neq (h'_{j^*}, s'_{j^*})$. From the two valid transcripts, $\mathcal{B}$ can reconstruct $sk = x_{Q+1}$ using the special soundness of the Schnorr identification scheme. Furthermore, since $(R'_j, h'_j, s'_j) = (X_j, h'_j, s'_j)$ is a valid transcript and x_{Q+1} is known, $\mathcal{B}$ can compute $x_j = s'_j - x_{Q+1} h'_j$ for all $j \in [Q]$. Finally, $\mathcal{B}$ returns $(x_1, \ldots, x_{Q+1})$, breaks OMDL problem with $\varepsilon = \varepsilon'$ and $t \approx t'$. ■

We now define the interactive discrete-logarithm problem which models PIMP-KOA-security for $\mathsf{ID_S}$.

Definition 5.6 (Q**-IDLOG**)**.** *The interactive discrete-logarithm assumption* Q-IDLOG *is said to be* (t, ε)*-hard in* $\mathsf{par} = (p, g, \mathbb{G})$ *if for all adversaries* $\mathcal{A}$ *running in time at most* t *and making at most* Q *queries to the challenge oracle* $\textsc{Ch}$*,*

$$\Pr\left[\; s \in \{x h_i + r_i \mid i \in [Q]\} \;\middle|\; \begin{array}{l} x \xleftarrow{\$} \mathbb{Z}_p; X = g^x \\ s \xleftarrow{\$} \mathcal{A}^{\textsc{Ch}(\cdot)}(X) \end{array} \right] \leq \varepsilon,$$

where on the i*-th query* $\textsc{Ch}(g^{r_i})$ ($i \in [Q]$)*, the challenge oracle returns* $h_i \xleftarrow{\$} \mathbb{Z}_p$ *to* $\mathcal{A}$*.*

In Appendix A we prove that in the generic group model, the Q-IDLOG problem in groups of prime-order p is at least $(2t^2/p, t)$-hard. Note that the bound is independent of Q.

5.1.2 Schnorr's Signature scheme

Let $H : \{0,1\}^* \to \{0,1\}^n$ be a hash function with $n < \log_2(p)$. As $\mathsf{ID_S}$ is commitment-recoverable we can use the alternative Fiat-Shamir transformation to obtain the Schnorr signature scheme $\mathsf{Schnorr} := (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$.

```
Gen(par):
$sk := x \xleftarrow{\$} \mathbb{Z}_p$
$pk := X = g^x$
Return $(pk, sk)$
```

```
Sign($sk, m$):
$r \xleftarrow{\$} \mathbb{Z}_p$; $R = g^r$
$h = H(R, m)$
$s = x \cdot h + r \bmod p$
$\sigma = (h, s) \in \{0,1\}^n \times \mathbb{Z}_p$
Return $\sigma$
```

```
Ver($sk, m, \sigma$):
Parse $\sigma = (h, s) \in \{0,1\}^n \times \mathbb{Z}_p$
$R = g^s X^{-h}$
If $h = H(R, m)$ then return 1
Else return 0.
```

By Theorem 3.1 and the results from this section, we obtain concrete bounds for Schnorr's single-user and multi-user security.

Lemma 5.7. *If* DLOG *is* (t, ε)*-hard in* $\mathsf{par} = (p, g, \mathbb{G})$ *then* $\mathsf{Schnorr}$ *is* $(t', \varepsilon', Q_s, Q_h)$-$\mathsf{SUF\text{-}CMA}$ *secure and* $(t'', \varepsilon'', N, Q_s, Q_h)$-$\mathsf{MU\text{-}SUF\text{-}CMA}$ *secure in the programmable random oracle model, where*

$$\begin{aligned} \frac{\varepsilon'}{t'} &\leq 6(Q_h + 1) \cdot \frac{\varepsilon}{t} + \frac{Q_s}{p} + \frac{1}{2^n}, \\ \frac{\varepsilon''}{t''} &\leq 24(Q_h + 1) \cdot \frac{\varepsilon}{t} + \frac{Q_s}{p} + \frac{1}{2^n}, \end{aligned}$$

The DLOG problem is tightly equivalent to the 1-IDLOG problem by Lemma 3.4. Assuming the OMDL problem is hard, $\mathsf{Schnorr}$ is $\mathsf{wIMP\text{-}MIM}$-secure and by Corollary 4.4 there cannot exist a tight implication 1-$\mathsf{IDLOG} \to Q$-IDLOG. Furthermore, by Theorem 3.2, the Q-IDLOG problem is tightly equivalent to $\mathsf{MU\text{-}SUF\text{-}CMA}$-security of $\mathsf{Schnorr}$.

Lemma 5.8. *If* Q_h-IDLOG *is* (t, ε)*-hard in* par *then* $\mathsf{Schnorr}$ *is* $(t', \varepsilon', N, Q_s, Q_h)$-$\mathsf{MU\text{-}SUF\text{-}CMA}$ *secure in the programmable random oracle model, where*

$$\varepsilon' \leq 4\varepsilon + \frac{Q_h Q_s}{p}, \qquad t' \approx t.$$

We leave it an open problem to come up with a more natural hard problem over par that tightly implies Q-IDLOG (and hence $\mathsf{MU\text{-}SUF\text{-}CMA}$-security of $\mathsf{Schnorr}$). Note that according to [FJS14], the hard problem has to have at least one round of interaction.

5.1.3 Concrete parameters

In this section we derive parameters for $\mathsf{Schnorr}$ providing k-bit security in the multi-user setting. Following [BR09], for k-bit security one requires $(\varepsilon', t', N, Q_s, Q_h)$-$\mathsf{MU\text{-}SUF\text{-}CMA}$ security with $\varepsilon'/t' \leq 2^{-k}$.

The following lemma assumes that a generic algorithm (for example, the Pollard-rho algorithm) is the best possible algorithm to break discrete logarithms in group $\mathbb{G}$. This is generally believed to be true for prime-order subgroups of elliptic curves.

Lemma 5.9. *Let* $\mathsf{Schnorr}$ *be instantiated with* $\mathsf{par} = (p, g, \mathbb{G}, H)$*, where* p *is a prime and* $H : \{0,1\}^* \to \{0,1\}^n$*. If a generic algorithm is the best possible algorithm to break discrete logarithms in group* $\mathbb{G}$*, then* $\mathsf{Schnorr}$ *provides* k*-bits security in the multi-user setting if*

$$\log p \geq 2k + \log(Q_h) + c'_{\mathsf{dl}}, \quad n \geq k + 1,$$

where c'_{dl} *is a constant that only depends on the generic algorithm. Furthermore, if a generic algorithm is the best possible algorithm to break the* Q-IDLOG *problem in* par*, then* $\mathsf{Schnorr}$ *provides* k*-bits security in the multi-user setting if*

$$\log p \geq 2k + c''_{\mathsf{dl}},$$

where c''_{dl} *is a constant that only depends on the generic algorithm.*

Proof. Assuming a generic algorithm is the best possible algorithm to compute discrete logarithms, means that DLOG in group $\mathbb{G}$ of prime-order p is $(\varepsilon = c_{\mathsf{dl}} \cdot t^2/p, t)$-hard, for any time bound t, where c_{dl} is a fixed constant that only depends on the specific choice of the generic algorithm.

We assume that the adversary makes $Q_h > 3$ hash queries. Define the constant $c'_{\mathsf{dl}} := 6 + \log(c_{\mathsf{dl}})$. Plugging in the parameters from Lemma 5.7 and using $Q_s \leq t \leq 2^k$ we obtain

$$\begin{aligned} \frac{\varepsilon'}{t'} &\leq 24(Q_h + 1)\frac{\varepsilon}{t} + \frac{Q_s}{p} + \frac{1}{2^n} \\ &\leq 32 Q_h c_{\mathsf{dl}} \frac{t}{p} + \frac{1}{2^n} \\ &\leq \frac{t}{2^{2k+1}} + \frac{1}{2^{k+1}} \leq 2^{-k} \end{aligned}$$

which proves the first part of the statement.

A similar computation can be done to prove the second part using Theorem A.1 saying that the best generic algorithm against Q-IDLOG has a success ratio of at most $\frac{2t^2}{p}$ ■

The interpretation for the multi-user security of Schnorr over elliptic-curve groups is as follows. It is well-known that a group of order p providing k-bits security against the DLOG problem requires $\log p \geq 2k$. If one requires provable security guarantees for Schnorr under DLOG, then one has to increase the group size by $\approx \log(Q_h)$ bits. Reasonable upper bounds for $\log Q_h$ are between 40 and 80. However, the generic lower bound of Theorem A.1 indicates that the only way to attack Schnorr in the sense of UF-KOA (and hence to attack Q-IDLOG) is to break the DLOG problem. In that case using groups with $\log p \approx 2k$ already gives provable security guarantees for Schnorr.

5.2 Katz-Wang Identification/Signature Scheme

5.2.1 Katz-Wang Identification Scheme

Let $\mathsf{par} := (p, g_1, g_2, \mathbb{G})$ be a set of system parameters, where $\mathbb{G} = \langle g_1 \rangle = \langle g_2 \rangle$ is a cyclic group of prime order p. The Katz-Wang identification scheme $\mathsf{ID}_{\mathsf{KW}} := (\mathsf{IGen}, \mathsf{P}, \mathsf{ChSet}, \mathsf{V})$ is defined as follows.

```
IGen(par):
  $sk := x \xleftarrow{\$} \mathbb{Z}_p$
  $pk := (X_1, X_2) = (g_1^x, g_2^x)$
  $\mathsf{ChSet} := \{0,1\}^n$
  Return $(pk, sk)$

V($pk, R = (R_1, R_2), h, s$):
  If $R_1 = g^s \cdot X_1^{-h}$ and $R_2 = g^s \cdot X_2^{-h}$ then return 1
  Else return 0

P_1($sk$):
  $r \xleftarrow{\$} \mathbb{Z}_p$; $R = (R_1, R_2) = (g_1^r, g_2^r)$
  $St := r$
  Return $(R, St)$

P_2($sk, R, h, St$):
  Parse $St = r$
  Return $s = x \cdot h + r \bmod p$
```

We recall the DDH assumption.

Definition 5.10 (Decision Diffie-Hellman Assumption). *The Decision Diffie-Hellman problem* DDH *is (t, ε)-hard in* $\mathsf{par} = (p, g_1, g_2, \mathbb{G})$ *if for all adversaries $\mathcal{A}$ running in time at most t,*

$$|\Pr\left[1 \xleftarrow{\$} \mathcal{A}(g_1^x, g_2^x) \mid x \xleftarrow{\$} \mathbb{Z}_p\right] - \Pr\left[1 \xleftarrow{\$} \mathcal{A}(g_1^{x_1}, g_2^{x_2}) \mid x_1 \xleftarrow{\$} \mathbb{Z}_p; x_2 \xleftarrow{\$} \mathbb{Z}_p \setminus \{x_1\}\right]| \leq \varepsilon.$$

Clearly, all security results of Schnorr carry over to the Katz-Wang identification scheme, i.e., $\mathsf{ID}_{\mathsf{KW}}$ is at least as secure as ID_{S}. That also means that we cannot hope for tight PIMP-KOA security from the DLOG assumption. Instead, for the Katz-Wang identification scheme, we give a direct tight proof of PIMP-KOA security under the DDH assumption.

Lemma 5.11. $\mathsf{ID}_{\mathsf{KW}}$ *is a canonical identification scheme with $\alpha = \log p$ bit min-entropy and it is unique, has special soundness (*SS*), honest-verifier zero-knowledge (*HVZK*) and is random-self reducible (*RSR*). Moreover, if* DDH *is (t, ε)-hard in* $\mathsf{par} = (p, g_1, g_2, \mathbb{G})$ *then* $\mathsf{ID}_{\mathsf{KW}}$ *is $(t', \varepsilon', Q_{\mathrm{CH}})$-*PIMP-KOA *secure, where $t \approx t'$ and $\varepsilon \geq \varepsilon' - Q_{\mathrm{CH}}/2^n$.*

Proof. The proof of SS, HVZK, uniqueness, and RSR is the same as in ID_{S}.

To prove PIMP-KOA-security under DDH, let $\mathcal{A}$ be an adversary that $(t', \varepsilon', Q_{\mathrm{CH}})$-breaks PIMP-KOA-security. We build an adversary $\mathcal{B}$ against the (t, ε)-hardness of DDH as follows. Adversary $\mathcal{B}$ inputs (X_1, X_2) and defines $pk = (X_1, X_2)$. On the i-th challenge query $\mathrm{CH}(R_{i,1}, R_{i,2})$, it returns $h_i \xleftarrow{\$} \mathbb{Z}_p$. Eventually, $\mathcal{A}$ returns $i^* \in [Q_{\mathrm{CH}}]$ and s_{i^*} and terminates. Finally, $\mathcal{B}$ outputs $d := \mathsf{V}(pk, R_{i^*}, h_{i^*}, s_{i^*})$.

Analysis of $\mathcal{B}$. If $(X_1, X_2) = (g_1^x, g_2^x)$, then $\mathcal{B}$ perfectly simulates the PIMP-KOA game and hence $\Pr[d = 1 \mid (X_1, X_2) = (g_1^x, g_2^x)] = \varepsilon'$. If $(X_1, X_2) = (g_1^{x_1}, g_2^{x_2})$ with $x_1 \neq x_2$, then we claim that even a computationally unbounded $\mathcal{A}$ can only win with probability $Q_{\mathrm{CH}}/2^n$, i.e., $\Pr[d = 1 \mid (X_1, X_2) = (g_1^{x_1}, g_2^{x_2})] \leq Q_{\mathrm{CH}}/2^n$.

It remains to prove the claim. For each index $i \in [Q_{\mathrm{CH}}]$, $\mathcal{A}$ first commits to $R_{i,1} = g_1^{r_{i,1}}$ and $R_{i,2} = g_2^{r_{i,2}}$ (for arbitrary $r_{i,1}, r_{i,2} \in \mathbb{Z}_p$) and can only win if there exists an $s_i \in \mathbb{Z}_p$ such that

$$\begin{aligned} & r_{i,1} + h_i x_1 = s_i = r_{i,2} + h_i x_2 \\ \Leftrightarrow \quad & h_i = \frac{r_{i,2} - r_{i,1}}{x_1 - x_2} \end{aligned}$$

where $h_i \xleftarrow{\$} \{0,1\}^n$ is chosen independently of $r_{i,1}, r_{i,2}$. This happens with probability at most $1/2^n$, so by the union bound we obtain the bound $Q_{\mathrm{CH}}/2^n$, as claimed. ■

5.2.2 Katz-Wang Signature scheme

Let $H : \{0,1\}^* \to \{0,1\}^n$ be a hash function with $n < \log_2(p)$. As $\mathsf{ID}_{\mathsf{KW}}$ is commitment-recoverable we can use the alternative Fiat-Shamir transformation to obtain the Katz-Wang signature scheme $\mathsf{KW} := (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$.

```
Gen(par):
    $sk := x \xleftarrow{\$} \mathbb{Z}_p$
    $pk := (X_1, X_2) = (g_1^x, g_2^x)$
    Return $(pk, sk)$
```

```
Sign(sk, m):
    $r \xleftarrow{\$} \mathbb{Z}_p$; $R = (R_1, R_2) = (g_1^{;} g_2^r)$
    $h = H(R, m)$
    $s = x \cdot h + r \bmod p$
    $\sigma = (h, s) \in \{0,1\}^n \times \mathbb{Z}_p$
    Return $\sigma$
```

```
Ver(sk, m, σ):
    Parse $\sigma = (h, s) \in \{0,1\}^n \times \mathbb{Z}_p$
    $R = g^s X^{-h}$
    If $h = H(R, m)$ then return 1
    Else return 0.
```

By our results we obtain the following concrete security statements, where the first bound matches [KW03, Theorem 1].

Lemma 5.12. *If* DDH *is* (t, ε)*-hard in* $\mathsf{par} = (p, g_1, g_2, \mathbb{G})$ *then* KW *is* $(t', \varepsilon', Q_s, Q_h)$-$\mathsf{SUF\text{-}CMA}$ *secure and* $(t'', \varepsilon'', N, Q_s, Q_h)$-$\mathsf{MU\text{-}SUF\text{-}CMA}$ *secure in the programmable random oracle model, where*

$$\begin{aligned} \frac{\varepsilon'}{t'} &\leq \frac{\varepsilon}{t} + \frac{Q_s}{p} + \frac{1}{2^n}, \\ \frac{\varepsilon''}{t''} &\leq 4 \cdot \frac{\varepsilon}{t} + \frac{Q_s}{p} + \frac{1}{2^n}. \end{aligned}$$

With a similar computation as in the case of $\mathsf{Schnorr}$, one can compute concrete parameters for k-bits security assuming that a generic algorithm is the best method to attack the DDH assumption in par. If $\log p \geq 2k + c'''_{\mathsf{dl}}$ and $n \geq k + 1$, then KW is $\mathsf{MU\text{-}SUF\text{-}CMA}$-secure, where c'''_{dl} is a constant that only depends on the generic algorithm.

5.3 Guillou-Quisquater Identification/Signature Scheme

5.3.1 Background on RSA and Notations

For $n \in \mathbb{N}$ we denote by $\mathbb{P}_{n/2}$ the set of all $n/2$-bit primes and $\mathsf{RSA}_n := \{(N = pq, p, q) \mid p, q \in \mathbb{P}_{n/2}, p \neq q\}$. Let $\phi(N) := (p-1)(q-1)$ be Euler's totient function for $(N, p, q) \in \mathsf{RSA}_n$. Let $\mathcal{R}$ be a relation on p and q. By $\mathsf{RSA}_n[\mathcal{R}]$ we denote the subset of RSA_n for that the relation $\mathcal{R}$ holds on p and q.

Let $n \in \mathbb{N}$ and $0 < c < \frac{1}{4}$ be a constant. We define the following two distributions.

$$\begin{aligned} \mathcal{I}_{n,c} &:= \{(N, e) \mid e \xleftarrow{\$} \mathbb{P}_{cn}; (N, p, q) \xleftarrow{\$} \mathsf{RSA}_n[\gcd(e, \phi(N) = 1)]\} \\ \mathcal{L}_{n,c} &:= \{(N, e) \mid e \xleftarrow{\$} \mathbb{P}_{cn}; (N, p, q) \xleftarrow{\$} \mathsf{RSA}_n[p = 1 \bmod e, p \neq 1 \bmod e^2, q \neq 1 \bmod e]\} \end{aligned}$$

Using the above notation, we recall the Phi-hiding assumption [CMS99, KOS10, KK12].

Definition 5.13 (Phi-hiding Assumption). *The Phi-hiding problem* ϕ-$\mathsf{H}_{n,c}$ *is* (t, ε)*-hard if for all adversaries* $\mathcal{A}$ *running in time at most* t*,*

$$|\Pr[1 \xleftarrow{\$} \mathcal{A}(N, e) \mid (N, e) \xleftarrow{\$} \mathcal{I}_{n,c}] - \Pr[1 \xleftarrow{\$} \mathcal{A}(N, e) \mid (N, e) \xleftarrow{\$} \mathcal{L}_{n,c}]| \leq \varepsilon.$$

5.3.2 Guillou-Quisquater Identification Scheme

Let $\mathsf{par} = (N, e) \xleftarrow{\$} \mathcal{I}_{n,c}$ be system parameters. The Guillou-Quisquater identification scheme $\mathsf{ID}_{\mathsf{GQ}} := (\mathsf{IGen}, \mathsf{P}, \mathsf{ChSet}, \mathsf{V})$ is defined as follows, where $\mathbb{Z}_N^* := \{y \in \mathbb{Z}_N \mid \gcd(y, N) = 1\}$.

```
IGen(par):
    $sk := x \xleftarrow{\$} \mathbb{Z}_N^*$
    $pk := X := x^e \bmod N$
    $\mathsf{ChSet} := \mathbb{Z}_e$
    Return $(pk, sk)$

V$(pk, R, h, s)$:
    If $R = s^e \cdot X^{-h} \bmod N$ and $((R, s) \in \mathbb{Z}_N^* \times \mathbb{Z}_N^*)$
    then return 1
    Else return 0
```

```
P$_1(sk)$:
    $r \xleftarrow{\$} \mathbb{Z}_N^*$; $R = r^e \bmod N$
    $St := r$
    Return $(R, St)$

P$_2(sk, R, h, St)$:
    Parse $St = r$
    Return $s = x^h \cdot r \bmod N$
```

It is easy to prove IMP-KOA security of $\mathsf{ID}_{\mathsf{GQ}}$ under the standard RSA assumption. Using our framework this implies MU-UF-CMA security of the implies GQ signature scheme, with an unavoidable security loss of Q_h. Under the ϕ-$\mathsf{H}_{n,c}$ assumption we can, however, give a direct tight proof of PIMP-KOA security, which is similar to [ABP13].

Lemma 5.14. *$\mathsf{ID}_{\mathsf{GQ}}$ is a canonical identification scheme with $\alpha = \log(\phi(N))$ bit min-entropy and it is unique, has special sound (*SS*), honest-verifier zero-knowledge (*HVZK*) and random-self reducible (*RSR*). Moreover, if ϕ-$\mathsf{H}_{n,c}$ is (t, ε)-hard then $\mathsf{ID}_{\mathsf{GQ}}$ is $(t', \varepsilon', Q_{\text{CH}})$-*PIMP-KOA *secure, where $t \approx t'$ and $\varepsilon \geq \varepsilon' - (Q_{\text{CH}} + 1)/e \geq \varepsilon' - (Q_{\text{CH}} + 1)/2^{cn}$.*

Proof. The correctness of $\mathsf{ID}_{\mathsf{GQ}}$ is straightforward to verify. We note that $R \xleftarrow{\$} \mathsf{P}_1(sk)$ is uniformly random over $\mathbb{Z}_N^*$. Hence, $\mathsf{ID}_{\mathsf{GQ}}$ has $\log|\mathbb{Z}_N^*| = \log(\phi(N))$ bit min-entropy. We show the other properties as follows.

Uniqueness. For all $(X, x) \in \mathsf{IGen}(\mathsf{par})$, $R := r^e \in \mathsf{P}_1(sk)$ and $h \in \mathbb{Z}_e$, the value $s \in \mathbb{Z}_N^*$ satisfying $s^e = X^h R \bmod N \Leftrightarrow s = x^h \cdot r \bmod N$ is uniquely defined, since $\gcd(e, \phi(N)) = 1$.

Special Soundness (SS). Given two accepting transcripts (R, h, s) and (R, h', s') with $h \neq h'$ (wlog. let $h > h'$), we have $s^e X^h = R = s'^e X^{h'} \bmod N$ and $(s/s')^e = X^{h-h'} \bmod N$. Since $h, h' \in \mathbb{Z}_e$, $\gcd(e, h - h') = 1$. Applying the extended Euclidean algorithm we can compute $A, B \in \mathbb{Z}_N^*$ such that

$$Ae + B(h - h') = \gcd(e, h - h') = 1.$$

Then we define an extractor algorithm $\mathsf{Ext}(X, R, h, s, h', s') := x^* := X^A (s/s')^B$ such that, for all $(X := x^e \bmod N, x) \in \mathsf{IGen}(\mathsf{par})$, we have $\Pr[(x^*)^e = X \bmod N] = 1$, since we have $(x^*)^e = (X^A(s/s')^B)^e = X^{Ae}(s/s')^{Be} = X^{Ae}X^{B(h-h')} = X$.

Honest-verifier Zero-knowledge (HVZK). Given a public key $pk = X$, we let $\mathsf{Sim}(pk)$ first sample $h \xleftarrow{\$} \mathbb{Z}_e$ and $s \xleftarrow{\$} \mathbb{Z}_N^*$ and then output $(R := s^e X^{-h} \bmod N, h, s)$. Clearly, (R, h, s) is a real transcript, since (h, s) is uniformly random over $\mathbb{Z}_e \times \mathbb{Z}_N^*$ and R is the unique value satisfying $R = s^e X^{-h} \bmod N$.

Random-self Reducibility (RSR). Algorithm Rerand and two deterministic algorithm Derand and Tran are defined as follows:

- $\mathsf{Rerand}(X_1)$ chooses $\mathbf{a}_2 \xleftarrow{\$} \mathbb{Z}_N^*$, computes $X_2 := X_1 \cdot \mathbf{a}_2{}^e \bmod N$ and returns $(X_2, \mathbf{a}_2)$. We have that, for all $(X_1, x_1) \in \mathsf{IGen}(\mathsf{par})$, X_2 is uniform and has the same distribution as X_3, where $(X_2, \mathbf{a}_2) \xleftarrow{\$} \mathsf{Rerand}(X_1)$ and $(X_3, x_3) \xleftarrow{\$} \mathsf{IGen}(\mathsf{par})$.
- $\mathsf{Derand}(X_1, X_2, x_2, \mathbf{a}_2)$ outputs $x^* = x_2/\mathbf{a}_2 \bmod N$. We have, for all $(X_2, \mathbf{a}_2) \xleftarrow{\$} \mathsf{Rerand}(X_1 := x_1^e \bmod N)$ with $(X_2, x_2) \in \mathsf{IGen}(\mathsf{par})$, $X_2 = x_2^e \bmod N$ and $x_2 = x_1 \cdot \mathbf{a}_2 \bmod N$ and thus $x^* = x_1 \bmod N$.
- $\mathsf{Tran}(X_1, X_2, \mathbf{a}_2, (R_2 := r_2^e \bmod N, h_2, s_2))$ outputs $s_1 = s_2/\mathbf{a}_2^{h_2} \bmod N$. We have, for all $(X_2, \mathbf{a}_2) \in \mathsf{Rerand}(X_1 := x_1^e \bmod N)$, if (R_2, h_2, s_2) is valid with respect to $X_2 := (x_1 \cdot \mathbf{a}_2)^e \bmod N$ then $s_1 = s_2/\mathbf{a}_2^{h_2} = (x_1\mathbf{a}_2)^{h_2} \cdot r_2/\mathbf{a}_2^{h_2} = x_1^{h_2} \cdot r_2 \bmod N$ and (R_2, h_2, s_1) is valid with respect to X_1.

PIMP-KOA Security. Let $\mathcal{A}$ be an adversary that $(t', \varepsilon', Q_{\text{CH}})$-breaks PIMP-KOA-security. We build an adversary $\mathcal{B}$ against the (t, ε)-hardness of ϕ-$\mathsf{H}_{n,c}$ as follows. Adversary $\mathcal{B}$ inputs (N, e), chooses $X \xleftarrow{\$} \mathbb{Z}_N^*$ and defines $pk := X$. On the i-th challenge query $\text{Ch}(R_i)$, it returns $h_i \xleftarrow{\$} \mathbb{Z}_e$. Eventually, $\mathcal{A}$ returns $i^* \in [Q_{\text{CH}}]$ and s_{i^*} and terminates. Finally, $\mathcal{B}$ outputs $d := \mathsf{V}(pk, R_{i^*}, h_{i^*}, s_{i^*})$.

Analysis of $\mathcal{B}$. If $(N, e) \xleftarrow{\$} \mathcal{I}_{n,c}$, then $\mathcal{B}$ perfectly simulates the PIMP-KOA game and hence $\Pr[d = 1 \mid (N, e) \xleftarrow{\$} \mathcal{I}_{n,c}] = \varepsilon'$. If $(N, e) \xleftarrow{\$} \mathcal{L}_{n,c}$ with $\gcd(N, e) \neq 1$, then we claim that even a computationally unbounded $\mathcal{A}$ can only win with probability $(Q_{\text{CH}} + 1)/e$, i.e., $\Pr[d = 1 \mid (N, e) \xleftarrow{\$} \mathcal{L}_{n,c}] \leq (Q_{\text{CH}} + 1)/e$.

It remains to prove the claim. Let $\mathcal{R}_e := \{X \mid \exists x \in \mathbb{Z}_N^* : X = x^e \bmod N\}$ be the set of all e-th residues in $\mathbb{Z}_N^*$. For $X, R \in \mathbb{Z}_N^*$ we first analyze

$$p(X,R) := \Pr_{h \xleftarrow{\boxdot} \mathbb{Z}_e} [\exists s \in \mathbb{Z}_N^* : \; s^e = X^h \cdot R \bmod N].$$

- Case 1: $X \in \mathcal{R}_e$. Then $p(X,R) \leq 1$ by choosing $s := (X^{1/e})^h \cdot R^{1/e}$ if $R \in \mathcal{R}_e$.
- Case 2: $X \notin \mathcal{R}_e$. Then $p(X,R) \leq 1/e$ because

$$\begin{aligned} p(X,R)(p(X,R)-1/e) &= \Pr_{h \neq \hat{h}} [\exists s, \hat{s} : s^e = X^h \cdot R \bmod N \wedge \hat{s}^e = X^{\hat{h}} \cdot R \bmod N] \\ &= \Pr_{h \neq \hat{h}} [\exists s, \hat{s} : (s/\hat{s})^e = X^{h-\hat{h}} \bmod N] = 0. \end{aligned}$$

The last equality holds, since $\gcd(e, h-\hat{h}) = 1$ which implies that $(s/\hat{s})^{e/(h-\hat{h})}$ is an e-th residue and cannot equal to $X \notin \mathcal{R}_e$.

Using the bounds on $p(X,R)$ we obtain

$$\begin{aligned} \Pr[\mathcal{A} \text{ wins}] &= \Pr[\mathcal{A} \text{ wins} \mid X \in \mathcal{R}_e] \Pr[X \in \mathcal{R}_e] + \Pr[\mathcal{A} \text{ wins} \mid X \notin \mathcal{R}_e] \cdot \Pr[X \notin \mathcal{R}_e] \\ &\leq \frac{1}{e} + (1 - \frac{1}{e}) \Pr_{h_1,\ldots,h_{Q_{\mathrm{CH}}}} [\bigvee \exists s_i : s_i^e = X^{h_i} \cdot R_i \bmod N \mid X \notin \mathcal{R}_e] \\ &\leq \frac{1}{e} + (1 - \frac{1}{e}) \frac{Q_{\mathrm{CH}}}{e} \\ &\leq \frac{Q_{\mathrm{CH}} + 1}{e} \end{aligned}$$

This completes the proof of the claim. ■

5.3.3 Guillou-Quisquater Signature Scheme

Let $H : \{0,1\}^* \to \mathbb{Z}_e$ be a hash function. As $\mathsf{ID}_{\mathsf{GQ}}$ is commitment-recoverable we can use the alternative Fiat-Shamir transformation to obtain the Guillou-Quisquater signature scheme $\mathsf{GQ} := (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$.

$\mathsf{Gen}(\mathsf{par})$:	$\mathsf{Sign}(sk, m)$:	$\mathsf{Ver}(sk, m, \sigma)$:
$sk := x \xleftarrow{\boxdot} \mathbb{Z}_N^*$ $X := x^e \bmod N$ $pk := X$ Return (pk, sk)	$r \xleftarrow{\boxdot} \mathbb{Z}_N^*$; $R = r^e \bmod N$ $h = H(R, m)$ $s = x^h \cdot r \bmod N$ $\sigma = (h, s) \in \mathbb{Z}_e \times \mathbb{Z}_N^*$ Return σ	Parse $\sigma = (h, s) \in \mathbb{Z}_e \times \mathbb{Z}_N^*$ $R = s^e X^{-h} \bmod N$ If $h = H(R, m)$ and $R \in \mathbb{Z}_N^*$ then return 1 Else return 0.

By our results we obtain the following concrete security statements.

Lemma 5.15. *If ϕ-$\mathsf{H}_{n,c}$ is (t, ε)-hard then* GQ *is $(t', \varepsilon', Q_s, Q_h)$-*$\mathsf{SUF\text{-}CMA}$ *secure and $(t'', \varepsilon'', N, Q_s, Q_h)$-*$\mathsf{MU\text{-}SUF\text{-}CMA}$ *secure in the programmable random oracle model, where*

$$\begin{aligned} \frac{\varepsilon'}{t'} &\leq \frac{\varepsilon}{t} + \frac{Q_s}{2^{n-2}} + \frac{3}{2^{cn}}, \\ \frac{\varepsilon''}{t''} &\leq 4 \cdot \frac{\varepsilon}{t} + \frac{Q_s}{2^{n-2}} + \frac{3}{2^{cn}}. \end{aligned}$$

References

[AABN02] Michel Abdalla, Jee Hea An, Mihir Bellare, and Chanathip Namprempre. From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 418–433. Springer, Heidelberg, April / May 2002. (Cited on page 1, 2, 4.)

[ABP13] Michel Abdalla, Fabrice Ben Hamouda, and David Pointcheval. Tighter reductions for forward-secure signature schemes. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, pages 292–311. Springer, Heidelberg, February / March 2013. (Cited on page 4, 24.)

[AFLT12] Michel Abdalla, Pierre-Alain Fouque, Vadim Lyubashevsky, and Mehdi Tibouchi. Tightly-secure signatures from lossy identification schemes. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 572–590. Springer, Heidelberg, April 2012. (Cited on page 4.)

[BDL$^+$11] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. High-speed high-security signatures. In Bart Preneel and Tsuyoshi Takagi, editors, *CHES 2011*, volume 6917 of *LNCS*, pages 124–142. Springer, Heidelberg, September / October 2011. (Cited on page 5.)

[Ber15a] Daniel Bernstein. [Cfrg] key as message prefix => multi-key security. https://mailarchive.ietf.org/arch/msg/cfrg/44gJyZlZ7-myJqWkChhpEF1KE9M, 2015. (Cited on page 5.)

[Ber15b] Daniel J. Bernstein. Multi-user Schnorr security, revisited. Cryptology ePrint Archive, Report 2015/996, 2015. http://eprint.iacr.org/. (Cited on page 5.)

[Bet88] Thomas Beth. Efficient zero-knowledge identification scheme for smart cards. In C. G. Günther, editor, *EUROCRYPT'88*, volume 330 of *LNCS*, pages 77–84. Springer, Heidelberg, May 1988. (Cited on page 1.)

[BGLS03] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 416–432. Springer, Heidelberg, May 2003. (Cited on page 5.)

[BM91] Ernest F. Brickell and Kevin S. McCurley. An interactive identification scheme based on discrete logarithms and factoring. In Ivan Damgård, editor, *EUROCRYPT'90*, volume 473 of *LNCS*, pages 63–71. Springer, Heidelberg, May 1991. (Cited on page 1.)

[BN06] Mihir Bellare and Gregory Neven. Multi-signatures in the plain public-key model and a general forking lemma. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 06*, pages 390–399. ACM Press, October / November 2006. (Cited on page 9.)

[BNPS03] Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. *Journal of Cryptology*, 16(3):185–215, June 2003. (Cited on page 19.)

[BP02] Mihir Bellare and Adriana Palacio. GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 162–177. Springer, Heidelberg, August 2002. (Cited on page 4, 8, 9, 10, 18, 19.)

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73. ACM Press, November 1993. (Cited on page 2, 6.)

[BR09] Mihir Bellare and Thomas Ristenpart. Simulation without the artificial abort: Simplified proof and improved concrete security for Waters' IBE scheme. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 407–424. Springer, Heidelberg, April 2009. (Cited on page 2, 21.)

[Bro15] Dan Brown. [Cfrg] key as message prefix => multi-key security. http://www.ietf.org/mail-archive/web/cfrg/current/msg07336.html, 2015. (Cited on page 5.)

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *30th ACM STOC*, pages 209–218. ACM Press, May 1998. (Cited on page 2.)

[CMS99] Christian Cachin, Silvio Micali, and Markus Stadler. Computationally private information retrieval with polylogarithmic communication. In Jacques Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 402–414. Springer, Heidelberg, May 1999. (Cited on page 23.)

[FF13] Marc Fischlin and Nils Fleischhacker. Limitations of the meta-reduction technique: The case of Schnorr signatures. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 444–460. Springer, Heidelberg, May 2013. (Cited on page 5.)

[FH15] Masayuki Fukumitsu and Shingo Hasegawa. Black-box separations on Fiat-Shamir-type signatures in the non-programmable random oracle model. In Javier Lopez and Chris J. Mitchell, editors, *ISC 2015*, volume 9290 of *LNCS*, pages 3–20. Springer, Heidelberg, September 2015. (Cited on page 3, 5.)

[FJS14] Nils Fleischhacker, Tibor Jager, and Dominique Schröder. On tight security proofs for Schnorr signatures. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 512–531. Springer, Heidelberg, December 2014. (Cited on page 5, 21.)

[FLR$^+$10] Marc Fischlin, Anja Lehmann, Thomas Ristenpart, Thomas Shrimpton, Martijn Stam, and Stefano Tessaro. Random oracles with(out) programmability. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 303–320. Springer, Heidelberg, December 2010. (Cited on page 2.)

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, August 1987. (Cited on page 1.)

[GBL08] Sanjam Garg, Raghav Bhaskar, and Satyanarayana V. Lokam. Improved bounds on security reductions for discrete log based signatures. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 93–107. Springer, Heidelberg, August 2008. (Cited on page 5.)

[Gir91] Marc Girault. An identity-based identification scheme based on discrete logarithms modulo a composite number (rump session). In Ivan Damgård, editor, *EUROCRYPT'90*, volume 473 of *LNCS*, pages 481–486. Springer, Heidelberg, May 1991. (Cited on page 1.)

[GJKW07] Eu-Jin Goh, Stanislaw Jarecki, Jonathan Katz, and Nan Wang. Efficient signature schemes with tight reductions to the Diffie-Hellman problems. *Journal of Cryptology*, 20(4):493–514, October 2007. (Cited on page 1, 4.)

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, April 1988. (Cited on page 1.)

[GMS02] Steven D. Galbraith, John Malone-Lee, and Nigel P. Smart. Public key signatures in the multi-user setting. *Inf. Process. Lett.*, 83(5):263–266, 2002. (Cited on page 1, 3, 5, 12.)

[GQ88] Louis C. Guillou and Jean-Jacques Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both trasmission and memory. In C. G. Günther, editor, *EUROCRYPT'88*, volume 330 of *LNCS*, pages 123–128. Springer, Heidelberg, May 1988. (Cited on page 4.)

[GQ90] Louis C. Guillou and Jean-Jacques Quisquater. A "paradoxical" indentity-based signature scheme resulting from zero-knowledge. In Shafi Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 216–231. Springer, Heidelberg, August 1990. (Cited on page 1, 18.)

[Ham15] Mike Hamburg. Re: [Cfrg] EC signature: next steps. `https://mailarchive.ietf.org/arch/msg/cfrg/af170b60rLyNZUHBMOPWxcDrVRI`, 2015. (Cited on page 5.)

[JL] S. Josefsson and I. Liusvaara. Edwards-curve digital signature algorithm (EdDSA), October 7, 2015. https://tools.ietf.org/html/draft-irtf-cfrg-eddsa-00. (Cited on page 5.)

[KK12] Saqib A. Kakvi and Eike Kiltz. Optimal security proofs for full domain hash, revisited. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 537–553. Springer, Heidelberg, April 2012. (Cited on page 23.)

[KOS10] Eike Kiltz, Adam O'Neill, and Adam Smith. Instantiability of RSA-OAEP under chosen-plaintext attack. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 295–313. Springer, Heidelberg, August 2010. (Cited on page 23.)

[KW03] Jonathan Katz and Nan Wang. Efficiency improvements for signature schemes with tight security reductions. In Sushil Jajodia, Vijayalakshmi Atluri, and Trent Jaeger, editors, *ACM CCS 03*, pages 155–164. ACM Press, October 2003. (Cited on page 1, 4, 18, 23.)

[Mau05] Ueli M. Maurer. Abstract models of computation in cryptography (invited paper). In Nigel P. Smart, editor, *10th IMA International Conference on Cryptography and Coding*, volume 3796 of *LNCS*, pages 1–12. Springer, Heidelberg, December 2005. (Cited on page 29.)

[MR02] Silvio Micali and Leonid Reyzin. Improving the exact security of digital signature schemes. *Journal of Cryptology*, 15(1):1–18, 2002. (Cited on page 9.)

[MS90] Silvio Micali and Adi Shamir. An improvement of the Fiat-Shamir identification and signature scheme. In Shafi Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 244–247. Springer, Heidelberg, August 1990. (Cited on page 1.)

[Oka93] Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 31–53. Springer, Heidelberg, August 1993. (Cited on page 1, 4.)

[OO98] Kazuo Ohta and Tatsuaki Okamoto. On concrete security treatment of signatures derived from identification. In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 354–369. Springer, Heidelberg, August 1998. (Cited on page 2, 4, 9.)

[OS91] H. Ong and Claus-Peter Schnorr. Fast signature generation with a Fiat-Shamir-like scheme. In Ivan Damgård, editor, *EUROCRYPT'90*, volume 473 of *LNCS*, pages 432–440. Springer, Heidelberg, May 1991. (Cited on page 1.)

[PS00] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000. (Cited on page 2, 3, 4, 9.)

[PV05] Pascal Paillier and Damien Vergnaud. Discrete-log-based signatures may not be equivalent to discrete log. In Bimal K. Roy, editor, *ASIACRYPT 2005*, volume 3788 of *LNCS*, pages 1–20. Springer, Heidelberg, December 2005. (Cited on page 3, 5.)

[Sch91] Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991. (Cited on page 1, 4, 18.)

[Seu12] Yannick Seurin. On the exact security of Schnorr-type signatures in the random oracle model. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 554–571. Springer, Heidelberg, April 2012. (Cited on page 3, 5, 9, 14, 16.)

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, Heidelberg, May 1997. (Cited on page 29.)

[Str15] Rene Struik. Re: [Cfrg] EC signature: next steps. https://mailarchive.ietf.org/arch/msg/cfrg/TOWH1DSzB-PfDGK8qEXtF3iC6Vc, 2015. (Cited on page 5.)

A Hardness of Q-IDLOG in the Generic Group Model

In the generic group model for the discrete logarithm setting [Sho97, Mau05], group operations in group $\mathbb{G}$ can only be carried out via an oracle $\mathcal{O}_{\mathbb{G}}$. Since $(\mathbb{G}, \cdot)$ of order p is isomorphic to $(\mathbb{Z}_p, +)$, elements from $\mathbb{G}$ are internally identified with elements from $\mathbb{Z}_p$. The oracle maintains a list that initially contains the elements $(1, C_1 = 1)$ (the generator), and $(x, C_x = 2)$ for $x \xleftarrow{\$} \mathbb{Z}_p$, and a counter i that counts the number of entries in the list and is initialized to 2. During the execution of the experiment, the list contains entries of the form (a, C_a), where $a \in \mathbb{Z}_p$ and $C_a \in \mathbb{N}$ is a counter. On input of two counters $C_a, C_b \in [c] \times [c]$, the oracle looks up the internal values (a, C_a) and (b, C_b), and computes $z = a + b$. If there already exists a tuple (z, C_z) in the list, then counter C_z is output. Otherwise, the counter i is increased by 1, the tuple $(z, C_z := i)$ is stored in the list, and the counter C_z is output.

Theorem A.1. *Let $\mathbb{G}$ be a group of prime order p. Then, in the generic group model, Q-IDLOG is (t, ε)-hard where*

$$\varepsilon \leq \frac{(Q_{\mathbb{G}} + 2)^2}{2p} + \frac{2Q}{p} \leq \frac{2t^2}{p},$$

and $Q_{\mathbb{G}}$ is the amount of queries to $\mathcal{O}_{\mathbb{G}}$.

Proof. Let $\mathcal{A}$ be an adversary against Q-IDLOG in the generic group model. In the proof we will simulate the list with entries of the form $(z(\mathbf{x}), C_{z(\mathbf{x})})$, where z is a polynomial of degree one in some variable $\mathbf{x}$. As we will see, our simulation will sometimes fail. Initially, the counter is set to $i = 2$ and the list contains the elements $(1, C_1 = 1)$ and $(\mathbf{x}, C_{\mathbf{x}} = 2)$, where $\mathbf{x}$ is a variable. After $\mathcal{A}$ has finished its execution, $\mathbf{x}$ will be assigned a value $x \xleftarrow{\$} \mathbb{Z}_p$. $\mathcal{A}$ is invoked on input $C_1 = 1$ and $C_{\mathbf{x}} = 2$. During its execution, $\mathcal{A}$ can query oracle $\mathcal{O}_{\mathbb{G}}$ on $(C_{a(\mathbf{x})}, C_{b(\mathbf{x})}) \in [i] \times [i]$. $\mathcal{O}_{\mathbb{G}}$ first computes the polynomial $z(\mathbf{x}) = a(\mathbf{x}) + b(\mathbf{x})$. If $(z(\mathbf{x}), C_{z(\mathbf{x})})$ is not in the list, $\mathcal{O}_{\mathbb{G}}$ increments counter i and adds $(z(\mathbf{x}), C_{z(\mathbf{x})} := i)$ to the list. Finally, $\mathcal{O}_{\mathbb{G}}$ outputs $C_{z(\mathbf{x})}$. In total, $\mathcal{A}$ makes $Q_{\mathbb{G}}$ queries to this oracle and we denote by $(z_i(\mathbf{x}), i)$ the i-entry in the list ($i \in [Q_{\mathbb{G}} + 2]$).

Furthermore, $\mathcal{A}$ can make queries to $\textsc{Ch}(j)$, for some counter $j \in [c]$, which is answered with $h_j \xleftarrow{\$} \mathbb{Z}_p$. For $j \in [Q]$, we denote by $(r_j(\mathbf{x}) = a_j\mathbf{x} + b_j, C_{r_j(\mathbf{x})})$ the polynomial associated to the j-th query to the $\textsc{Ch}$ oracle. Eventually, $\mathcal{A}$ outputs $s \in \mathbb{Z}_p$ and terminates. Next, $x \xleftarrow{\$} \mathbb{Z}_p$ is chosen and $\mathcal{A}$ wins if there is a $j \in [Q]$ such that $s = (h_j + a_j)x + b_j$.

We remark that we simulate the $\mathcal{O}_{\mathbb{G}}$ perfectly, if none of the distinct polynomials $z_i(\mathbf{x})$ collide when evaluated on input x. We define Bad as the event that this is the case, i.e. there exist an $i \neq \ell \in [Q_{\mathbb{G}}]$ such that the polynomials $z_i(\mathbf{x})$, $z_\ell(\mathbf{x})$ are distinct but $z_i(x) = z_\ell(x)$. By a union bound we first bound

$$\begin{aligned}
\Pr[\mathrm{Bad}] &= \Pr_x[(\exists i, \ell \in [Q_{\mathbb{G}}] \times [Q_{\mathbb{G}}] : z_i(\mathbf{x}) \neq z_\ell(\mathbf{x}) \wedge z_i(x) = z_\ell(x)] \\
&\leq \binom{Q_{\mathbb{G}} + 2}{2} \frac{1}{p} \leq \frac{(Q_{\mathbb{G}} + 2)^2}{2p}.
\end{aligned}$$

The success probability ε of $\mathcal{A}$ can be bounded as

$$\begin{aligned}
\varepsilon &\leq \Pr[\mathrm{Bad} \vee \exists j \in [Q] : s = (h_j + a_j)x + b_j] \\
&\leq \Pr[\mathrm{Bad}] + \Pr[\exists j \in [Q] : s = (h_j + a_j)x + b_j] \\
&\leq \frac{(Q_{\mathbb{G}} + 2)^2}{2p} + \Pr_x[\exists j \in [Q] : s = (h_j + a_j)x + b_j \mid h_j \neq -a_j] + \Pr_{h_1, \ldots, h_Q}[\exists j \in [Q] : h_j = -a_j] \\
&\leq \frac{(Q_{\mathbb{G}} + 2)^2}{2p} + \frac{2Q}{p}
\end{aligned}$$

This completes the proof. ∎

B From Single-user to multi-user security for Schnorr

The following result provides an alternative way to prove (a slightly tighter version of) Theorem 3.2 for the special case of Schnorr signatures. It proves that SUF-CMA security tightly implies MU-SUF-CMA security in the standard model. Note that we require strong security but using Lemma 3.7, UF-KOA security tightly implies SUF-CMA (and hence MU-SUF-CMA) security of Schnorr in the random oracle model.

Lemma B.1 (SUF-CMA ⇒ MU-SUF-CMA)**.** *If* Schnorr *is* (t, ε, Q_s)-SUF-CMA *secure then, for any* $N \geq 1$, Schnorr *is* $(t', \varepsilon', N, Q_s)$-MU-SUF-CMA *secure, where*

$$\varepsilon' \leq 2\varepsilon + \frac{Q_s^2}{p}, \quad t' \approx t,$$

Q_s is an upper bounds on the number of signing queries and N is the number of users.

Proof. Let $\mathcal{A}$ be an adversary that breaks $(t', \varepsilon', N, Q_s)$-MU-SUF-CMA security of Schnorr. We construct an adversary $\mathcal{B}$ that breaks (t, ε, Q_s)-SUF-CMA security of Schnorr. Adversary $\mathcal{B}$ is executed in the SUF-CMA experiment. It obtains a public-key $pk = X = g^x$ and has access to a signing oracle $\textsc{Sign}$.

$\textsc{Simulation of public-keys}$. First, for each $i \in [N]$, adversary $\mathcal{B}$ picks $a_i \overset{\$}{\leftarrow} \mathbb{Z}_p$, secret bits $b_i \overset{\$}{\leftarrow} \{0, 1\}$, and computes

$$pk_i = X_i := X^{b_i} \cdot g^{a_i}. \tag{9}$$

That is, if $b_i = 0$, then $sk_i = a_i$ is known to $\mathcal{B}$; if $b_i = 1$ then $sk_i = x + a_i$ is unknown to $\mathcal{B}$. Note that the public-keys are correctly distributed. Next, $\mathcal{B}$ runs $\mathcal{A}$ on input $(pk_1, \dots, pk_N)$ answering signing queries as follows.

$\textsc{Simulation of signing queries}$. On $\mathcal{A}$'s j-th signing query $(i_j, m_j) \in [N] \times \{0, 1\}^*$, $\mathcal{B}$ is supposed to return a signature σ_j on message m_j under pk_{i_j}. Those are computed by adversary $\mathcal{B}$ according to the following case distinction.

- Case A: $b_{i_j} = 0$. In that case $sk_{i_j} = a_{i_j}$ is known to $\mathcal{B}$ and the signature is computed as $\sigma_j := (h_j, s_j) \overset{\$}{\leftarrow} \mathsf{Sign}(sk_{i_j}, m_j)$.
- Case B: $b_{i_j} = 1$. In that case $sk_{i_j} = x + a_{i_j}$ is unknown to $\mathcal{B}$ and the signature is computed using $\mathcal{B}$'s signing oracle by first querying $(h_j, \hat{s}_j) \overset{\$}{\leftarrow} \textsc{Sign}(m_j)$. Then $\sigma_j = (h_j, s_j := \hat{s}_j + a_{i_j} h_j)$ is a valid signature on message m_j under pk_{i_j}. Indeed, $\mathsf{Ver}(pk_{i_j}, m_j) = 1$ because $H(g^{s_j} X_{i_j}^{-h_j}, m_j) = H(g^{\hat{s}_j} X^{-h_j}, m_j) = h_j$.

Adversary $\mathcal{B}$ returns $\sigma_j = (h_j, s_j)$ which in both cases is a correctly distributed valid signature. For future reference we also define $R_j := g^{s_j} X_{i_j}^{-h_j}$ and by (9)

$$r_j := \log_g(R_j) = s_j - (b_{i_j} x + a_{i_j}) h_j. \tag{10}$$

We assume that

$$\forall k \neq j \in [Q_s] : \quad r_k \neq r_j. \tag{11}$$

Since s_j and hence r_j are uniform elements from $\mathbb{Z}_p$, condition (11) is not satisfied with probability at most Q_s^2/p. Note that the simulation of the public-keys and the signing queries do not leak any information about the secret bits b_i.

$\textsc{Forgery}$. Eventually, $\mathcal{A}$ will submit a forgery $(i^*, m^*, \sigma^* := (h^*, s^*))$ and terminate. For the remainder of this proof we assume σ^* is a correct signature on m^* under pk_{i^*}, i.e., for $R^* := g^{s^*} X_{i^*}^{-h^*}$ it holds that $H(R^*, m^*) = h^*$. Using (9) the correctness condition can be equivalently expressed as

$$r^* := \log_g(R^*) = s^* - (b_{i^*} x + a_{i^*}) h^*. \tag{12}$$

Furthermore we assume that σ^* is a valid fresh forgery in the MU-SUF-CMA experiment:

$$(i^*, m^*, h^*, s^*) \notin \{(i_j, m_j, h_j, s_j) \mid j \in [Q_s]\}. \tag{13}$$

After receiving $\mathcal{A}$'s forgery, $\mathcal{B}$ is supposed to compute its own valid forgery under $pk = X$. To this end, $\mathcal{B}$ defines the set of all indices j such that it queried m_j to its signing oracle $\mathcal{J} := \{j \in [Q_s] \mid b_{i_j} = 1\}$ and makes the following case distinction. A pictorial overview of all cases is given in Figure 5.

- Case 1: For all $j \in [Q_s]$ we have: $h^* \neq h_j$ or $r^* \neq r_j$,

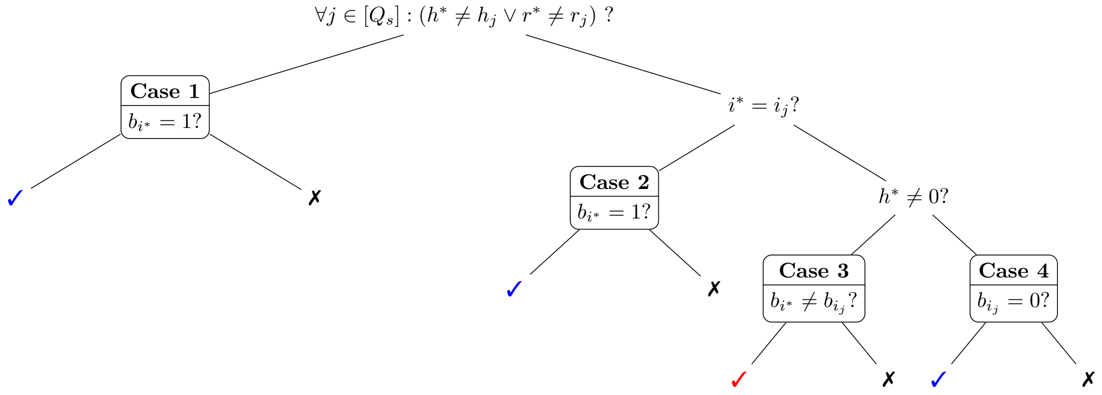


Figure 5: Overview of the case distinction in the proof of Lemma B.1. Each node contains a condition. If the condition is satisfied then we continue to the left child, otherwise to the right child. A leaf denotes either a good case (getting a valid SUF-CMA forgery, marked with "✓", or extracting the secret-key, marked with "✓") or a bad case, marked with "✗" (in which we abort).

  - Case 1a: $b_{i^*} = 1$. Then for $\hat{s}^* := s^* - a_{i^*}h^*$ we have

$$H(g^{\hat{s}^*}X^{-h^*}, m^*) = H(g^{s^*}X_{i^*}^{-h^*}, m^*) = h^*$$

    and hence

$$\hat{\sigma}^* := (h^*, \hat{s}^*)$$

    is a correct signature on message m^* under $pk = X$. It remains to show that $\hat{\sigma}^*$ is a fresh strong forgery in the SUF-CMA experiment.

    On the one hand, if $h^* \notin \{h_1, \dots, h_{Q_s}\}$, we directly obtain $\hat{\sigma}^* = (h^*, \hat{s}^*) \notin \{(h_j, \hat{s}_j) \mid j \in \mathcal{J}\}$ (the set of all signatures obtained from the SUF-CMA signing oracle) which means that $(m^*, \hat{\sigma}^*)$ satisfies the freshness condition of the SUF-CMA experiment. On the other hand, if the set $\mathcal{J}^*$ of indices $j \in [Q_s]$ such that $h_j = h^*$ is non-empty, then we will use the condition $r^* \neq r_j$ to show that the corresponding $\hat{s}_j$ values are all distinct from $\hat{s}^*$. Indeed, for all $k \in \mathcal{J}^* \cap \mathcal{J}$ we have $\hat{s}_k = r_k + xh^* \neq r^* + xh^*$ and therefore $\hat{s}^* = r^* + xh^* \notin \{\hat{s}_k \mid k \in \mathcal{J}^* \cap \mathcal{J}\}$. For all $k \in \mathcal{J} \setminus \mathcal{J}^*$ we have $h^* \neq h_k$ and therefore $h^* \notin \{h_k \mid k \in \mathcal{J} \setminus \mathcal{J}^*\}$. Consequently, $\hat{\sigma}^* = (h^*, \hat{s}^*) \notin \{(h_k, \hat{s}_k) \mid k \in \mathcal{J}\}$ and $(m^*, \hat{\sigma}^*)$ satisfies the freshness condition of the SUF-CMA experiment.

  - Case 1b: $b_{i^*} = 0$. Then $\mathcal{B}$ aborts.

  Note that in case 1, $\mathcal{B}$ aborts with probability exactly 1/2. If it does not abort, it outputs a valid strong forgery.

- Case 2: There exists a $j \in [Q_s]$ such that $h^* = h_j$ and $r^* = r_j$ and $i^* = i_j$.

  Note that if j exists it is uniquely defined by (11).

  - Case 2a: $b_{i^*} = 1$. As in case 1a,

$$\hat{\sigma}^* := (h^*, \hat{s}^* := s^* - a_{i^*}h^*)$$

    is a correct signature on message m^* under $pk = X$. By $r^* = r_j$ and $h^* = h_j$ we obtain $(h^*, s^*) = (h_j, s_j)$. Since we also have $i^* = i_j$, $\mathcal{A}$'s freshness condition (13) implies $m^* \neq m_j$ meaning that $\hat{\sigma}^*$ is a valid fresh forgery in the SUF-CMA experiment.

  - Case 2b: $b_{i^*} = 0$. Then $\mathcal{B}$ aborts.

  Note that in case 2, $\mathcal{B}$ aborts with probability exactly 1/2. If it does not abort, it outputs a valid strong forgery.

- Case 3: There exists a $j \in [Q_s]$ such that $h^* = h_j \neq 0$ and $r^* = r_j$ and $i^* \neq i_j$.

  Note that if j exists it is uniquely defined by (11).

  - Case 3a: $b_{i_j} \neq b_{i^*}$. By (10) and (12) we obtain two equations in the intermediates (r^*, x)

    $$\begin{aligned} r^* &= s^* - (b_{i^*} x + a_{i^*}) h^* \\ r^* &= s_j - (b_{i_j} x + a_{i_j}) h^*, \end{aligned}$$

    from which $\mathcal{B}$ can extract the single-user scheme's secret-key $x = \log_g(X)$ as

    $$x := ((s^* - s_j)(h^*)^{-1} + a_{i_j} - a_{i^*}) \cdot (b_{i^*} - b_{i_j})^{-1}.$$

    Using $sk = x$, $\mathcal{B}$ computes a valid forgery on any fresh message.

  - Case 3b: $b_{i_j} = b_{i^*}$. Then $\mathcal{B}$ aborts.

  Note that in case 3, since $b_{i^*} \neq b_{i_j}$, $\mathcal{B}$ aborts with probability exactly 1/2. If it does not abort, it outputs a valid strong forgery.

- Case 4: There exists a $j \in [Q_s]$ such that $h_j = h^* = 0$ and $r^* = r_j$ and $i^* \neq i_j$.[6]

  Again, if j exists it is uniquely defined by (11).

  - Case 4a: $b_{i_j} = 0$. Then

    $$\hat{\sigma}^* := (0, s^*)$$

    is a correct signature on m^* under $pk = X$. For all $k \neq j$ with $h_k = h^* = 0$ we have by (11) $r^* \neq r_k$ and therefore $s^* = r^* \neq r_k = \hat{s}_k$. This means that $\hat{\sigma}^* = (0, s^*) = (0, r^*) \notin \{(h_k, \hat{s}_k) \mid k \in \mathcal{J}\}$ (the set of all signatures obtained from the SUF-CMA signing oracle). Therefore $(m^*, \hat{\sigma}^*)$ satisfies the freshness condition of the SUF-CMA experiment.

  - Case 4b: $b_{i_j} = 1$. Then $\mathcal{B}$ aborts.

  Note that in case 4, $\mathcal{B}$ aborts with probability exactly 1/2. If it does not abort, it outputs a valid strong forgery.

Overall, $\mathcal{B}$ returns a fresh strong forgery $(m^*, \hat{\sigma}^*)$ under $pk = X$ with probability $\varepsilon = \frac{1}{2}\left(\varepsilon' - \frac{Q_s^2}{p}\right)$. Adversary $\mathcal{B}$ makes at most Q_s signing queries (in expectation only $Q_s/2$). Its running time is that of $\mathcal{A}$ plus some additional small computation for each signing query and each user (which we neglect), hence $t' \approx t$. ■

We remark that due to forgery cases 1 and 4 our reduction requires strong SUF-CMA security and does not work with standard UF-CMA security.

[6] By assuming the hash function to be zero-resistant we may as well discard this case.