# Schnorr Signatures in the Multi-User Setting

Eike Kiltz Daniel Masny Jiaxin Pan

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

**Abstract**

A theorem by Galbraith, Malone-Lee, and Smart (GMLS) from 2002 showed that, for Schnorr signatures, single-user security tightly implies multi-user security. Recently, Bernstein pointed to an error in the above theorem and promoted a key-prefixing variant of Schnorr signatures for which he proved a tight implication from single to multi-user security. Even worse, he identified an "apparently insurmountable obstacle to the claimed [GMLS] theorem".

This paper shows that, without key prefixing, single-user security of Schnorr signatures tightly implies multi-user security of the same scheme.

**Keywords:** Schnorr signatures, multi-user security, unforgeability, tight reduction

## 1 Introduction

Single-user vs. multi-user security for signature schemes. 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) [10] 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 $Q$ 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 attack (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 $Q$ 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 [8] 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 probability 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 as it loses 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 probability by a factor of $2^{30}$. For example, if we assume the best algorithm breaking the single-user security having success probability $\varepsilon = 2^{-80}$, then it can only be argued that the best algorithm breaking the multi-user security has success probability $\varepsilon' = 2^{-80} \cdot 2^{30} = 2^{-50}$, which is not a safe security margin that defends against today's attackers.

| Security notion | Multi-user setting? | Signing queries allowed? | Strong unforgeability? |
|---|---|---|---|
| UF-KOA | — | — | — |
| UF-CMA | — | √ | — |
| SUF-CMA | — | √ | √ |
| MU-UF-CMA | √ | √ | — |
| MU-SUF-CMA | √ | √ | √ |

**Figure 1:** Overview of the considered security notions for signature schemes. Here MU stands for multi-user, (S)UF stands for (strong) unforgeability, KOA stands for key only attack, and CMA stands for chosen message attack.

---

Schnorr signatures and their multi-user security. One of the most important signature schemes is the Schnorr signature scheme [15]. To mitigate the generic security loss problem discussed above for the special case of Schnorr's signature scheme, Galbraith, Malone-Lee, and Smart (GMLS) proved [8] 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 probability compared to attacking the scheme in the single-user setting. Unfortunately, Bernstein [3] 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 [3] further expands on the insurmountable obstacle.

## 1.1 Our results

Our main result states that the original GMLS theorem is correct, namely that single-user security of the Schnorr signatures scheme tightly implies multi-user security of the same scheme. We even show a much stronger result, namely that multi-user security of Schnorr's signature scheme is already tightly implied by the weak single-user security notion of unforgeability against key-only attack (UF-KOA). In UF-KOA security the adversary is only given the public-key and has to produce a valid forgery without obtaining any signature.

To state our results more formally, we consider the security notions described in Figure 1. In the multi-user setting, the adversary obtains $N$ public-keys, in the single-user setting exactly one. If signing queries are allowed, the adversary can ask for signatures on messages under any of the obtained public-key(s). He wins, if he outputs a valid signature/message pair that verifies on any of the obtained public-key(s). For standard unforgeability we exclude the trivial attacks, where the adversary forges for a message/public-key pair he queried the signing oracle on. For strong unforgeability we relax the winning condition and exclude only those attacks, where the adversary forges for a message/public-key/signature triple he previously obtained from the signing oracle. Note that strong and standard unforgeability is the same for key-only attacks.

The implications among the security notions for Schnorr signatures are shown in Figure 2. Theorem 3.1 (SUF-CMA $\rightarrow$ MU-SUF-CMA) is our main result and the intuition behind its proof is given below. We note that Theorem 3.2 (UF-KOA$\rightarrow$SUF-CMA in the random oracle model) is already contained implicitly in Pointcheval and Stern's early work on Schnorr signatures [14]. Its proof is given here for completeness. (Intuitively, it is true since the Schnorr identification protocol is honest-verifier zero-knowledge and hence the signatures can be simulated by programming the random oracle.) Combining these two theorems we recover the original GMLS theorem UF-CMA$\rightarrow$MU-SUF-CMA in the random oracle model. We leave it as an open problem to prove the GMLS theorem in the standard model.

To complete the picture of provable security, [14] also proved that the discrete logarithm (DLOG) assumption implies UF-KOA security of Schnorr's signature scheme in the random oracle model. However, their security reduction uses the Forking Lemma and therefore it not tight. Furthermore, a tight reduction (from UF-KOA security) to the DLOG assumption (or even any natural, possibly interactive, computational assumption) can be proven to be impossible [13, 9, 16, 7].

Overall, this gives a complete picture of the security of Schnorr signatures. Our interpretation is as follows. Strong security in the multi-user setting (MU-SUF-CMA) is tightly equivalent to UF-KOA security.

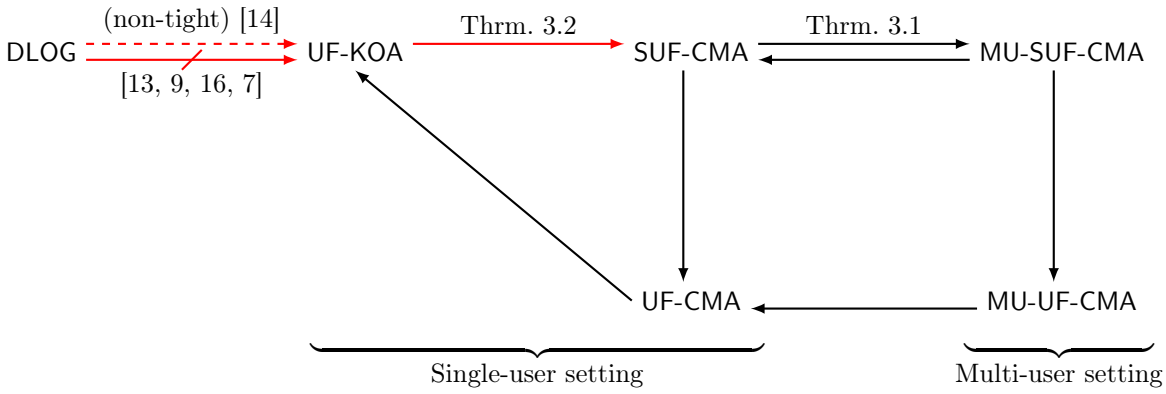


**Figure 2:** Implications among various security notions for the Schnorr signature scheme and the discrete logarithm assumption DLOG. Here $\mathsf{X} \to \mathsf{Y}$ means that X-security tightly implies Y-security and $\mathsf{X} \dashrightarrow \mathsf{Y}$ means that X-security loosely implies Y-security. Striked lines indicate impossibility results. Red arrows denote implications in the random oracle model and black arrows unconditional implications. All remaining implications can be derived from the diagram using transitivity. In particular, all notions are tightly equivalent in the random oracle model. For example, UF-CMA→MU-SUF-CMA is obtained via the path UF-CMA → UF-KOA→SUF-CMA → MU-SUF-CMA.

---

The latter is a simple non-interactive assumption defined over a cyclic group $\mathbb{G} = \langle g \rangle$, namely that given $g^x$ and hash function $H$, it is hard to come up with $(h, s)$ such that $H(g^{s-xh}) = h$. Even though this non-interactive assumption is not tightly equivalent to the standard DLOG assumption, still more than 25 years of cryptanalytic effort have failed to find an attack without breaking the DLOG assumption. Hence it is reasonable to assume Schnorr's UF-KOA security. In contrast, UF-CMA and MU-UF-CMA security are more complex interactive security notions which have not been the target of cryptanalytic efforts. Our results show that such efforts are useless.

Proof details of the main theorem. We give some details about the proof of Theorem 3.1 (SUF-CMA→MU-SUF-CMA). The basic idea of the original GMLS security reduction is that from a given public key $\mathsf{pk}$ we can derive properly distributed $\mathsf{pk}_1, \ldots, \mathsf{pk}_N$ such that any signature $\hat{\sigma}$ which is valid under $\mathsf{pk}$ can be transformed into a signature $\sigma$ which is valid under $\mathsf{pk}_i$ and vice-versa.

The transformation is used as an interface between the single and the multi-user setting. That is, in the reduction the multi-user signing queries on message $\mathsf{m}_i$ under $\mathsf{pk}_j$ can be perfectly simulated by single-user signing queries on message $\mathsf{m}_i$ under $\mathsf{pk}$. A forgery of the multi-user adversary is transformed back into a forgery in the single-user setting. Can we argue that this is a valid forgery? As pointed out by Bernstein [3], the problem in the GMLS reduction is that a multi-user adversary could first obtain a signature on message $\mathsf{m}$ under $\mathsf{pk}_1$ and then submit a valid forgery on the same message $\mathsf{m}$ but under $\mathsf{pk}_2$. In that case the above reduction fails to produce a valid forgery, since the reduction queries for a signature on message $\mathsf{m}$ and later submits a forgery on the *same message* $\mathsf{m}$.

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 derived using the above transformation, for the other half the reduction knows the corresponding secret-keys. Which keys are known is hidden from the adversary. Now, if the multi-user adversary first obtains a signature on message $\mathsf{m}$ under $\mathsf{pk}_1$ and then submits a forgery on the same message $\mathsf{m}$ under $\mathsf{pk}_2$, the reduction hopes for the good case that one of the public-keys was obtained using the transformation and the other one is known. This happens with probability $1/2$ which is precisely the loss of our new reduction. In the good case we either get a valid forgery for the single-user case or efficiently extract the secret key $\mathsf{sk}$ (similar to the extraction of the secret-key using the Forking Lemma). (If the multi-user adversary submits a forgery on a message he did not previously query the signing oracle on, we simply use the old GMLS reduction.)

### 1.2 Schnorr signatures vs. Key-Prefixed Schnorr signatures

After identifying the error in the GMLS proof, Bernstein [3] 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 [4] 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 [5].) In [4] key-prefixing is advertized as "an inexpensive way to alleviate concerns that several public keys could be attacked simultaneously." Indeed, Bernstein [3] proves that single-user security of the original Schnorr signatures scheme tightly implies multi-user security of the key-prefixed variant of the scheme.

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 [11] 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 [11] and Struik [17] (among others) recommended against key prefixing for Schnorr. Shortly after, Bernstein [2] 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" [6]. As a consequence, key-prefixing is contained in the current draft for EdDSA [12]. In the light of our new results, we recommend to reconsider this decision.

The history of tight security of standard Schnorr signatures in the multi-user setting also shows that provable security aspects *should* play (among other things) an integral role in security evaluations and deciding about future standards. In fact, our result is the consequence of a failed attempt to formally prove the impossibility of a tight reduction.

## 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 $a \leftarrow A$ denotes picking $a$ from $A$ according to the uniform distribution. All our algorithms are probabilistic unless states otherwise. If $\mathsf{A}$ is an algorithm, then $a \leftarrow \mathsf{A}(b)$ denotes the random variable which is defined as the output of $\mathsf{A}$ on input $b$.

### 2.2 Digital Signatures

We now define the syntax of a digital signature scheme. Let $\mathsf{par}$ be common system parameters shared among all participants.

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

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

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

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

$$\Pr\left[\begin{array}{l|l} \mathsf{Ver}(\mathsf{pk}_{i^*}, \mathsf{m}^*, \sigma^*) = 1 & \textit{For } i = 1, \ldots, N : (\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{Gen}(\mathsf{par}) \\ \wedge\ (i^*, \mathsf{m}^*, \sigma^*) \notin \{(i_j, \mathsf{m}_j, \sigma_j) \mid j \in [Q_s]\} & (i^*, \mathsf{m}^*, \sigma^*) \leftarrow \mathsf{A}^{\textsc{Sign}(\cdot,\cdot)}(\mathsf{pk}_1, \ldots, \mathsf{pk}_N) \end{array}\right] \leq \varepsilon,$$

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

We stress an adversary in particular breaks multi-user security if he asks for a signature on message $\mathsf{m}$ under $\mathsf{pk}_1$ and submits a valid forgery on the same message $\mathsf{m}$ under $\mathsf{pk}_2$.

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) MU-UF-CMA security (multi-user unforgeablility against chosen message attack) we modify the freshness condition in the experiment to $(i^*, \mathsf{m}^*) \notin \{(i_j, \mathsf{m}_j, ) \mid j \in [Q_s]\}$, 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.

**Definition 2.3** (Single-user security of signature schemes)**.** *In the single-user setting, i.e. $N = 1$ users, $(t, \varepsilon, Q_s)$-SUF-CMA security (strong unforgeablility against chosen message attack) is defined as $(t, \varepsilon, 1, Q_s)$-MU-SUF-CMA security. Similarly, standard (non-strong) $(t, \varepsilon, Q_s)$-UF-CMA security (unforgeablility against chosen message attack) is defined as $(t, \varepsilon, 1, Q_s)$-MU-UF-CMA security. Further, $(t, \varepsilon)$-UF-KOA security (unforgeablility against key-only attack) is defined as $(t, \varepsilon, 1, 0)$-MU-SUF-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 [1]. 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 $(t, \varepsilon, N, Q_s, Q_h)$-MU-SUF-CMA security we consider all adversaries making at most $Q_h$ queries to the random oracle.

# 3 Schnorr's Signature scheme

In this section let $\mathsf{par} := (H, p, g, \mathbb{G})$ 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 $\mathbb{Z}_q^*$. We assume that each element $x \in \mathbb{G}$ has a unique representation as a bit-string in $\{0,1\}^{\ell_\mathbb{G}}$, for some integer $\ell_\mathbb{G}$. Function $H : \{0,1\}^* \to \{0,1\}^n$ is a hash function with $n < \log_2(p)$. The Schnorr signature scheme $\mathsf{Schnorr} := (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ is defined as follows:

| $\underline{\mathsf{Gen}(\mathsf{par})}$: | $\underline{\mathsf{Sign}(\mathsf{sk}, \mathsf{m})}$: | $\underline{\mathsf{Ver}(\mathsf{sk}, \mathsf{m}, \sigma)}$: |
|---|---|---|
| $\mathsf{sk} := x \leftarrow \mathbb{Z}_p$ | $r \leftarrow \mathbb{Z}_p$; $R = g^r$ | Parse $\sigma = (h, s) \in \{0,1\}^n \times \mathbb{Z}_p$ |
| $\mathsf{pk} := X = g^x$ | $h = H(R, \mathsf{m})$ | $R = g^s X^{-h}$ |
| Return $(\mathsf{pk}, \mathsf{sk})$ | $s = x \cdot h + r \bmod p$ | If $h = H(R, \mathsf{m})$ then return 1 |
| | Return $\sigma = (h, s) \in \{0,1\}^n \times \mathbb{Z}_p$ | Else return 0. |

## 3.1 Single-user tightly implies multi-user security

The following result is our main theorem and says that SUF-CMA security tightly implies MU-SUF-CMA security of Schnorr's signature scheme. Our theorem only requires strong security but via the chain of implication from Figure 2 we obtain that UF-KOA security (and therefore in particular UF-CMA security) tightly implies MU-SUF-CMA security of Schnorr in the random oracle model.

**Theorem 3.1** (SUF-CMA $\Rightarrow$ MU-SUF-CMA). *If* Schnorr *is* $(t, \varepsilon, Q_s)$-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 $\mathsf{A}$ be an adversary that breaks $(t', \varepsilon', N, Q_s)$-MU-SUF-CMA security of Schnorr. We construct an adversary $\mathsf{B}$ that breaks $(t, \varepsilon, Q_s)$-SUF-CMA security of Schnorr. Adversary $\mathsf{B}$ is executed in the SUF-CMA experiment. It obtains a public-key $\mathsf{pk} = X = g^x$ and has access to a signing oracle Sign.

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

$$\mathsf{pk}_i = X_i := X^{b_i} \cdot g^{a_i}. \tag{1}$$

That is, if $b_i = 0$, then $\mathsf{sk}_i = a_i$ is known to $\mathsf{B}$; if $b_i = 1$ then $\mathsf{sk}_i = x + a_i$ is unknown to $\mathsf{B}$. Note that the public-keys are correctly distributed. Next, $\mathsf{B}$ runs $\mathsf{A}$ on input $(\mathsf{pk}_1, \ldots, \mathsf{pk}_N)$ answering signing queries as follows.

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

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

Adversary $\mathsf{B}$ 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 (1)

$$r_j := \log_g(R_j) = s_j - (b_{i_j} x + a_{i_j}) h_j. \tag{2}$$

We assume that

$$\forall k \neq j \in [Q_s] : \quad r_k \neq r_j. \tag{3}$$

Since $s_j$ and hence $r_j$ are uniform elements from $\mathbb{Z}_p$, condition (3) 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 $b_i$.

Forgery. Eventually, $\mathsf{A}$ will submit a forgery $(i^*, \mathsf{m}^*, \sigma^* := (h^*, s^*))$ and terminate. For the remainder of this proof we assume $\sigma^*$ is a correct signature on $\mathsf{m}^*$ under $\mathsf{pk}_{i^*}$, i.e., for $R^* := g^{s^*} X_{i^*}^{-h^*}$ it holds that $H(R^*, \mathsf{m}^*) = h^*$. Using (1) the correctness condition can be equivalently expressed as

$$r^* := \log_g(R^*) = s^* - (b_{i^*} x + a_{i^*}) h^*. \tag{4}$$

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

$$(i^*, \mathsf{m}^*, h^*, s^*) \notin \{(i_j, \mathsf{m}_j, h_j, s_j) \mid j \in [Q_s]\}. \tag{5}$$

After receiving $\mathsf{A}$'s forgery, $\mathsf{B}$ is supposed to compute its own valid forgery under $\mathsf{pk} = X$. To this end, $\mathsf{B}$ defines the set of all indices $j$ such that it queried $\mathsf{m}_j$ 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 3.

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

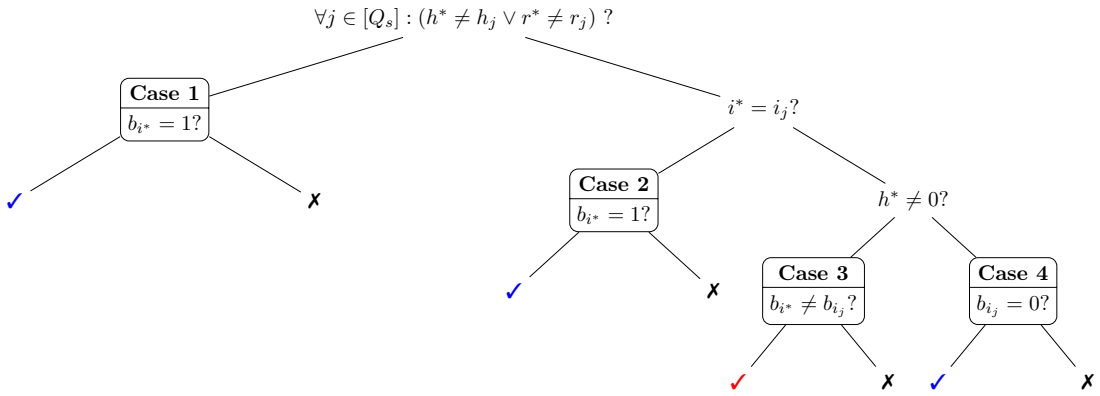


**Figure 3:** Overview of the case distinction in the proof of Theorem 3.1. Each node contains a condition. If the condition is satisfied then we continue to the left child, otherwise to the right child. A leaf denotes either a good case (getting a valid SUF-CMA forgery, marked with "✓", or extracting 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

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

and hence

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

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

On the one hand, if $h^* \notin \{h_1, \ldots, h_{Q_s}\}$, 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 $(\mathsf{m}^*, \hat{\sigma}^*)$ 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 $(\mathsf{m}^*, \hat{\sigma}^*)$ satisfies the freshness condition of the SUF-CMA experiment.

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

Note that in case 1, 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 (3).

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

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

  is a correct signature on message $\mathsf{m}^*$ under $\mathsf{pk} = X$. By $r^* = r_j$ and $h^* = h_j$ we obtain $(h^*, s^*) = (h_j, s_j)$. Since we also have $i^* = i_j$, A's freshness condition (5) implies $\mathsf{m}^* \neq \mathsf{m}_j$ meaning that $\hat{\sigma}^*$ is a valid fresh forgery in the SUF-CMA experiment.

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

Note that in case 2, 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 (3).
  - Case 3a: $b_{i_j} \neq b_{i^*}$. By (2) and (4) 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 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 $\mathsf{sk} = x$, B computes a valid forgery on any fresh message.
  - Case 3b: $b_{i_j} = b_{i^*}$. Then B aborts.

  Note that in case 3, since $b_{i^*} \neq b_{i_j}$, 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 $i^* \neq i_j$.*
  Again, if $j$ exists it is uniquely defined by (3).
  - Case 4a: $b_{i_j} = 0$. Then

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

    is a correct signature on $\mathsf{m}^*$ under $\mathsf{pk} = X$. For all $k \neq j$ with $h_k = h^* = 0$ we have by (3) $r^* \neq r_k$ 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 $(\mathsf{m}^*, \hat{\sigma}^*)$ satisfies the freshness condition of the SUF-CMA experiment.
  - Case 4b: $b_{i_j} = 1$. Then B aborts.

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

Overall, B returns a fresh strong forgery $(\mathsf{m}^*, \hat{\sigma}^*)$ under $\mathsf{pk} = X$ with probability $\varepsilon = \frac{1}{2}\left(\varepsilon' - \frac{Q_s^2}{p}\right)$. Adversary B makes at most $Q_s$ signing queries (in expectation only $Q_s/2$). Its running time is that of 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.

## 3.2 Key-only security tightly implies single-user security

The following result is implicitly contained in [14]. Its proof is given for completeness.

**Theorem 3.2** (UF-KOA ⇒ SUF-CMA)**.** *If* Schnorr *is* $(t, \varepsilon, Q_h)$-UF-KOA *secure and* $H$ *is modeled as a random oracle, then* Schnorr *is* $(t', \varepsilon', Q_s, Q_h)$-SUF-CMA *secure, where*

$$\varepsilon' \leq \varepsilon + \frac{(Q_h + Q_s)Q_s}{p}, \quad t' \approx t,$$

*and* $Q_s$*,* $Q_h$ *are upper bounds on the number of signing and hash queries, respectively.*

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

*Proof.* Let $\mathsf{A}$ be an algorithm that breaks $(t', \varepsilon', Q_s)$-SUF-CMA security of Schnorr. We will describe an adversary $\mathsf{B}$ invoking $\mathsf{A}$ that breaks $(t, \varepsilon)$-UF-KOA security of Schnorr with $(t, \varepsilon)$ as stated in the theorem. Adversary $\mathsf{B}$ is executed in the UF-KOA experiment and obtains a public-key $\mathsf{pk} := X = g^x$, and has access to a random oracle $H$. $\mathsf{B}$ runs $\mathsf{A}$ on input $\mathsf{pk}$ answering hash and signing queries as follows.

Simulation of hash queries. A hash query is answered by $\mathsf{B}$ by querying its own hash oracle and returning its answer.

Simulation of signing queries. On the $j$-th signature query on message $\mathsf{m}_j$, $\mathsf{B}$ samples random $s_j \leftarrow \mathbb{Z}_p$, $h_j \leftarrow \{0,1\}^n$ and computes $R_j := g^{s_j} \cdot X^{-h_j}$. If $H(R_j, \mathsf{m}_j)$ was already defined (via one of $\mathsf{A}$'s queries), $\mathsf{B}$ aborts. Otherwise, it defines the random oracle

$$H(R_j, \mathsf{m}_j) := h_j \tag{6}$$

and returns $\sigma_j := (h_j, s_j)$, which is a correctly distributed valid signatures on $\mathsf{m}_j$. Note that for each signing query $\mathsf{B}$ aborts with probability at most $(Q_h + Q_s)/p$ because $R_j$ is uniformly distributed. Since the number of signing queries is bounded by $Q_s$, $\mathsf{B}$ aborts overall with probability at most $(Q_h + Q_s)Q_s/p$.

Forgery. Eventually, $\mathsf{A}$ will submit its forgery $(\mathsf{m}^*, \sigma^* := (h^*, s^*))$. We assume that it is a valid forgery, i.e., for $R^* = g^{s^*} X^{-h^*}$ we have $H(R^*, \mathsf{m}^*) = h^*$ and

$$(\mathsf{m}^*, h^*, s^*) \notin \{(\mathsf{m}_j, h_j, s_j) : j \in [Q_s]\}. \tag{7}$$

If there exists a $j \in [Q_s]$ such that $(R^*, \mathsf{m}^*) = (R_j, \mathsf{m}_j)$ then we have $h^* = h_j$ and $s^* = s_j$ which contradicts the freshness condition (7). This implies $(R^*, \mathsf{m}^*) \notin \{(R_i, \mathsf{m}_i) : i \in [Q_s]\}$ which means that the hash value $H(R^*, \mathsf{m}^*)$ was not programmed by $\mathsf{B}$ in (6). Finally $\mathsf{B}$ returns $(\mathsf{m}^*, \sigma^*)$ to its UF-KOA experiment, which is a valid fresh forgery. Its running time is that of $\mathsf{A}$ plus some additional small computation for each signing query (which we neglect), hence $t' \approx t$. □

# References


[1] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73. ACM Press, Nov. 1993. (Cited on page 5.)

[2] D. Bernstein. [Cfrg] key as message prefix => multi-key security. `https://mailarchive.ietf.org/arch/msg/cfrg/44gJyZlZ7-myJqWkChhpEF1KE9M`, 2015. (Cited on page 4.)

[3] D. J. Bernstein. Multi-user Schnorr security, revisited. Cryptology ePrint Archive, Report 2015/996, 2015. `http://eprint.iacr.org/`. (Cited on page 2, 3, 4.)

[4] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang. High-speed high-security signatures. In B. Preneel and T. Takagi, editors, *CHES 2011*, volume 6917 of *LNCS*, pages 124–142. Springer, Heidelberg, Sept. / Oct. 2011. (Cited on page 4.)

[5] D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In E. Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 416–432. Springer, Heidelberg, May 2003. (Cited on page 4.)

[6] D. Brown. [Cfrg] key as message prefix => multi-key security. `http://www.ietf.org/mail-archive/web/cfrg/current/msg07336.html`, 2015. (Cited on page 4.)

[7] N. Fleischhacker, T. Jager, and D. Schröder. On tight security proofs for Schnorr signatures. In P. Sarkar and T. Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 512–531. Springer, Heidelberg, Dec. 2014. (Cited on page 2, 3.)

[8] S. D. Galbraith, J. Malone-Lee, and N. P. Smart. Public key signatures in the multi-user setting. *Inf. Process. Lett.*, 83(5):263–266, 2002. (Cited on page 1, 2.)

[9] S. Garg, R. Bhaskar, and S. V. Lokam. Improved bounds on security reductions for discrete log based signatures. In D. Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 93–107. Springer, Heidelberg, Aug. 2008. (Cited on page 2, 3.)

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

[11] M. Hamburg. Re: [Cfrg] EC signature: next steps. `https://mailarchive.ietf.org/arch/msg/cfrg/af170b6OrLyNZUHBMOPWxcDrVRI`, 2015. (Cited on page 4.)

[12] 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 4.)

[13] P. Paillier and D. Vergnaud. Discrete-log-based signatures may not be equivalent to discrete log. In B. K. Roy, editor, *ASIACRYPT 2005*, volume 3788 of *LNCS*, pages 1–20. Springer, Heidelberg, Dec. 2005. (Cited on page 2, 3.)

[14] D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000. (Cited on page 2, 3, 8.)

[15] C.-P. Schnorr. Efficient identification and signatures for smart cards. In G. Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 239–252. Springer, Heidelberg, Aug. 1990. (Cited on page 2.)

[16] Y. Seurin. On the exact security of schnorr-type signatures in the random oracle model. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 554–571. Springer, Heidelberg, Apr. 2012. (Cited on page 2, 3.)

[17] R. Struik. Re: [Cfrg] EC signature: next steps. `https://mailarchive.ietf.org/arch/msg/cfrg/TOWH1DSzB-PfDGK8qEXtF3iC6Vc`, 2015. (Cited on page 4.)