Cryptogr. Commun. (2018) 10:41–58
DOI 10.1007/s12095-017-0253-6

Security of BLS and BGLS signatures in a multi-user setting

Marie-Sarah Lacharité[1]

Received: 30 November 2016 / Accepted: 8 August 2017 / Published online: 24 August 2017

Abstract Traditional single-user security models do not necessarily capture the power of real-world attackers. A scheme that is secure in the single-user setting may not be as secure in the multi-user setting. Inspired by the recent analysis of Schnorr signatures in the multi-user setting, we analyse Boneh-Lynn-Shacham (BLS) signatures and Boneh-Gentry-Lynn-Shacham (BGLS) aggregate signatures in the multi-user setting. We obtain a tight reduction from the security of key-prefixed BLS in the multi-user model to normal BLS in the single-user model. We introduce a multi-user security model for general aggregate signature schemes, in contrast to the original "chosen-key" security model of BGLS that is analogous to the single-user setting of a signature scheme. We obtain a tight reduction from the security of multi-user key-prefixed BGLS to the security of multi-user key-prefixed BLS. Finally, we apply a technique of Katz and Wang to present a tight security reduction from a variant of multi-user key-prefixed BGLS to the computational co-Diffie-Hellman (co-CDH) problem. All of our results for BLS and BGLS use type III pairings.

Keywords Security models · Signatures · Aggregate signatures · Multi-user · Reduction

Mathematics Subject Classifications (2010) 94A60 · 68P25

1 Introduction

It is important to have security models that reflect real-world conditions. Proving that a scheme is secure against a particular type of adversary is less valuable when it does not correspond to the adversary of such a system deployed in the real world. While single-user

This article is part of the Topical Collection on *Recent Trends in Cryptography*

✉ Marie-Sarah Lacharité
marie-sarah.lacharite.2015@rhul.ac.uk

[1] Information Security Group, Royal Holloway, University of London, TW20 0EX, Egham, UK

settings are much easier to analyse, they do not capture many real-world attacks. For example, computing the greatest common divisors of RSA moduli from different public keys makes it possible to recover their corresponding private keys if these moduli share a prime factor [12]. A generic MAC scheme's security in the multi-user setting is not equivalent to its security in the single-user setting—a MAC forger in the multi-user setting has an advantage over a MAC forger in the single-user setting by a factor of the number of users, n [9].

Although digital signatures have existed for over 30 years, there is still debate about the most appropriate security models and how to interpret security reductions when choosing security parameters. As recently as late 2015, the security of Schnorr signatures in the multi-user setting relative to their security in the single-user setting was not well understood [4, 5, 15]. This discussion about the security of Schnorr signatures inspired our analysis of two more signature schemes.

1.1 Overview and our contributions

In this paper, we analyse the multi-user security of two related signature schemes: the Boneh-Lynn-Shacham (BLS) signature scheme [7] and the Boneh-Gentry-Lynn-Shacham (BGLS) aggregate signature scheme [6]. In Sections 1.2–1.4, we review Schnorr and BLS signatures, unforgeability, and single-user vs. multi-user settings. In Section 1.5, we summarize the history of multi-user Schnorr signature security and related work.

Next, in Section 2, we present a reduction from multi-user BLS security to single-user BLS security. It is not tight; the gap is about $\min\{n, q_s + 1\}$. We go on to show in Theorem 1 that with key-prefixing, the security of multi-user BLS *does* tightly reduce to the security of single-user BLS. Although we were unable to obtain a tight reduction from the security of multi-user BLS to single-user BLS without key-prefixing—which does not eliminate the possibility that multi-user forgery is easier—it is in fact known that, in the random oracle model, multi-user BLS has the same tightness loss as single-user BLS when it is reduced directly to the co-CDH problem [14]. We explain this in more detail at the end of Section 2.

In Section 3, we consider BGLS, a natural extension of BLS and the first proposed aggregate signature scheme. We introduce a truly multi-user security model for general aggregate signature schemes, as opposed to the chosen-key model of BGLS, in Section 3.1. In Theorem 2, we present a tight reduction from *key-prefixed* multi-user BGLS security to *key-prefixed* multi-user BLS security in the random-oracle model. Finally, in Theorem 3, we present a tight security reduction for the key-prefixed BGLS scheme in the multi-user setting, also in the random oracle model, by further modifying BGLS to use a technique of Katz and Wang [13].

Figure 1 summarizes our results and shows where they fit in relation to known results. The tightness of reductions is indicated with "tight" or with the tightness gap. "KP" indicates the variant where keys are prefixed to messages before signing, and "KW" indicates the Katz-Wang variants, where a random bit is prefixed to a message before signing. Reductions are in the standard model unless otherwise indicated with "RO" for "random oracle."

1.2 Review of Schnorr and BLS signatures, notation

See Table 1 for an overview of the Schnorr [19] and BLS [7] signature schemes. For simplicity, we omit the role of the security parameter in Setup. Schnorr signatures are mentioned in this paper for illustrative purposes; although we contribute no results about Schnorr signatures, the history of their security models inspired this work.

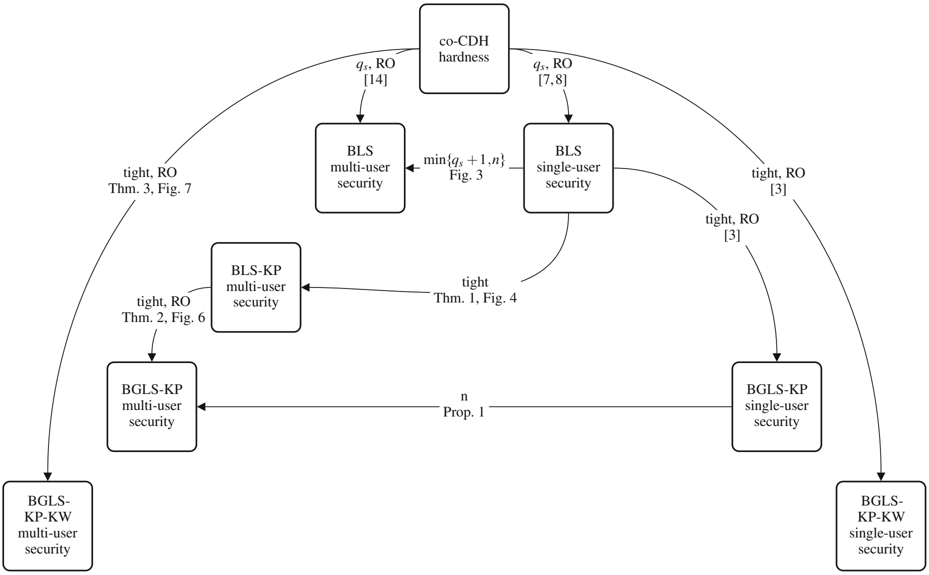


Fig. 1 Overview of our results for BLS and BGLS and how they compare to reductions in the single-user setting

Although the BLS scheme was introduced for symmetric pairings, where $G_1 = G_2$, we use the modified scheme due to Chatterjee et al. [8] that also works for asymmetric pairings where no efficiently computable isomorphism from G_2 to G_1 is known (“type III” pairings).

The security of the BLS signature scheme with type III pairings depends on the hardness of the **computational co-Diffie-Hellman (co-CDH) problem** in $G_1 \times G_2$: given $({g_1}^x, {g_2}^x) \in G_1 \times G_2$ and $h \in G_1$, compute $h^x \in G_1$. We say that a problem is **($\mathbf{t}$, ϵ)-hard** if there exists no adversary that can solve it in time at most t with probability at least ϵ, where the probability is taken over all possible instances of the problem and any coin flips the adversary makes.

Table 1 Summary of the Schnorr and BLS signature schemes

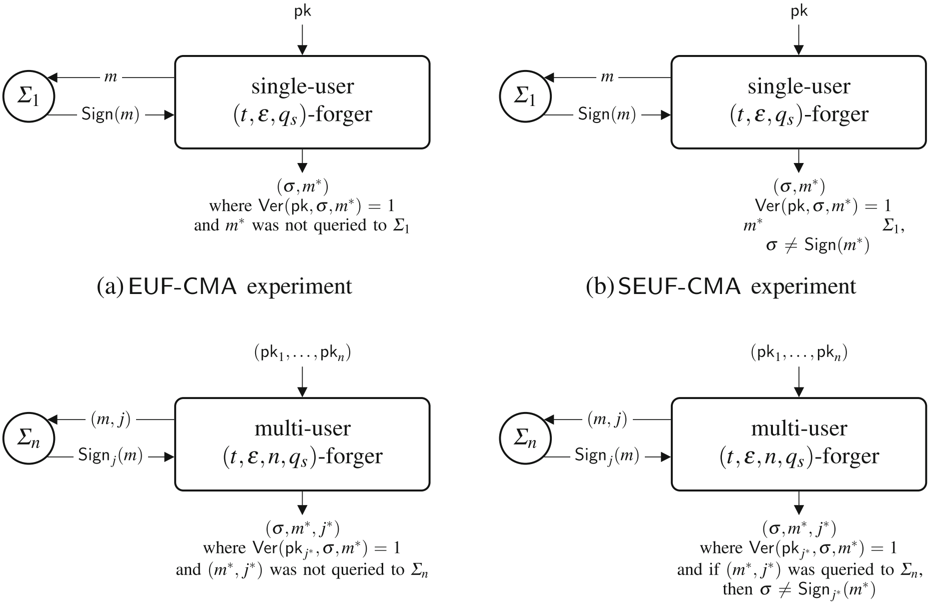
	Schnorr [19]	BLS [7]
Setup	$G = \langle g \rangle$, prime order p $\mathsf{H} : \{0, 1\}^* \to \mathbb{Z}_p$, full-domain	$G_1 = \langle g_1 \rangle$, $G_2 = \langle g_2 \rangle$, G_T, prime order p $\mathsf{H} : \{0, 1\}^* \to G_1$, full-domain $e : G_1 \times G_2 \to G_T$, bilinear
KeyGen	$x \leftarrow_\$ \mathbb{Z}_p$ $\mathsf{sk} = x \in \mathbb{Z}_p$ $\mathsf{pk} = g^x \in G$	$x \leftarrow_\$ \mathbb{Z}_p$ $\mathsf{sk} = x \in \mathbb{Z}_p$ $\mathsf{pk} = (y_1, y_2) = ({g_1}^x, {g_2}^x) \in G_1 \times G_2$
$\mathsf{Sign}(\mathsf{sk}, m)$	$k \leftarrow_\$ \mathbb{Z}_p$ $\sigma = (h, s)$ $= (\mathsf{H}(g^k \| m), \mathsf{sk} \cdot h + k)$ $\in \mathbb{Z}_p \times \mathbb{Z}_p$	$\sigma = \mathsf{H}(m)^{\mathsf{sk}} \in G_1$
$\mathsf{Ver}(\mathsf{pk}, \sigma, m)$	$h \overset{?}{=} \mathsf{H}(g^s \cdot \mathsf{pk}^{-h} \| m)$	$e(\mathsf{H}(m), y_2) \overset{?}{=} e(\sigma, g_2)$

Note that BLS public keys contain both ${g_1}^x$ and ${g_2}^x$, but only the latter is used for verification. Without ${g_1}^x$, it is unknown [8] whether there is a reduction from BLS forgery to solving the co-CDH problem (the opposite direction of that which we examine in this paper). If a forger receives a public key $({g_1}^x, {g_2}^x)$, it is straightforward to see that, given a co-CDH solver, it can forge a signature on any message without even making a single signing query.

We let t_m and t_e represent the times required to compute a multiplication or exponentiation in G_1 or G_2. We write $[n]$ for the set of integers $\{1, \ldots, n\}$. We write $x \leftarrow_\$ S$ to denote picking a value of x uniformly at random from the set S. We let $x||y$ denote the concatenation of (the binary representations of) x and y. A multi-set is a set that may contain repetitions.

1.3 Standard and strong unforgeability

The widely-accepted notion of security for a digital signature scheme is resistance to existential forgery under adaptive chosen-message attacks, formalized by Goldwasser, Micali, and Rivest in 1988 [11]. A digital signature scheme is (single-user) **($\mathbf{t}$, ϵ, $\mathbf{q_s}$)-existentially unforgeable under adaptive chosen-message attacks** (EUF-CMA) if there exists no forger $\mathscr{F}$ that, given one challenge public key generated by KeyGen and adaptively making at most q_s queries to a signing oracle, runs in time at most t and can produce with probability at least ϵ a signature on a message it did not submit to the signing oracle. See Fig. 2a for a diagram representing the EUF-CMA experiment.


pk
Σ_1
m
Sign(m)
single-user
(t, ε, q_s)-forger
(σ, m^*)
where Ver$(\mathsf{pk}, \sigma, m^*) = 1$
and m^* was not queried to Σ_1
pk
Σ_1
m
Sign(m)
single-user
(t, ε, q_s)-forger
(σ, m^*)
Ver$(\mathsf{pk}, \sigma, m^*) = 1$
m^* Σ_1,
$\sigma \neq$ Sign(m^*)
$(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$
Σ_n
(m, j)
Sign$_j(m)$
multi-user
(t, ε, n, q_s)-forger
(σ, m^*, j^*)
where Ver$(\mathsf{pk}_{j^*}, \sigma, m^*) = 1$
and (m^*, j^*) was not queried to Σ_n
$(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$
Σ_n
(m, j)
Sign$_j(m)$
multi-user
(t, ε, n, q_s)-forger
(σ, m^*, j^*)
where Ver$(\mathsf{pk}_{j^*}, \sigma, m^*) = 1$
and if (m^*, j^*) was queried to Σ_n,
then $\sigma \neq$ Sign$_{j^*}(m^*)$

(a) EUF-CMA experiment

(b) SEUF-CMA experiment

(c) MU-EUF-CMA experiment

(d) MU-SEUF-CMA experiment

Fig. 2 Four types of existential forgery experiments in the standard model. The public keys are generated with KeyGen. Similar experiments exist in the random oracle model, where the forgers can also make at most q_h queries to a hashing oracle

For probabilistic signature schemes, there exists a variant of existential unforgeability: "strong unforgeability," introduced by An, Dodis, and Rabin in 2002 [1]. A digital signature scheme is (single-user) ***strongly*** **($\mathbf{t}, \epsilon, \mathbf{q_s}$)-existentially unforgeable under adaptive chosen-message attacks** (SEUF-CMA) if there is no forger $\mathscr{F}$, with the same properties as above, that can produce a new signature on any message, including the messages it submitted to the signing oracle. (See Fig. 2b.)

Although the notions of standard and strong unforgeability are identical for BLS, the difference is important in understanding the history of the security models of Schnorr signatures.

1.4 Single-user and multi-user settings

In the standard and strong unforgeability models, the adversary receives one target public key for which it must forge a signature. However, since public keys are *public*, a real-world adversary has the choice of which public key to target. Perhaps it is easier to forge a signature for any public key from a set rather than one specific public key. That is, perhaps forgery in the multi-user setting, where the adversary chooses which public key to target, is easier than forgery in the single-user setting. We will compare these two settings.

The study of signature schemes in the multi-user setting was initiated by Menezes and Smart in 2001 [18]. A digital signature scheme is (multi-user) **($\mathbf{t}, \epsilon, \mathbf{n}, \mathbf{q_s}$)-existentially unforgeable under adaptive chosen-message attacks** (MU-EUF-CMA) if there exists no forger $\mathscr{F}$ that, given n challenge public keys generated by KeyGen and adaptively making at most q_s queries to a signing oracle, runs in time at most t and with probability at least ϵ, can produce a signature for one of the n users on a message that it did not submit to the signing oracle for this user. (See Fig. 2c.) For probabilistic signature schemes, there is again a strong unforgeability variant of multi-user security. A digital signature scheme is (multi-user) ***strongly*** **($\mathbf{t}, \epsilon, \mathbf{n}, \mathbf{q_s}$)-existentially unforgeable under adaptive chosen-message attacks** (MU-SEUF-CMA) if there is no forger $\mathscr{F}$, with the same properties as above, that can produce a new signature on any message by any of the users, including messages submitted to the signing oracle for this user. (See Fig. 2d.)

1.5 A brief history of multi-user schnorr signature security and related work

In this section, we review the security of Schnorr signatures in the multi-user model. We will employ similar techniques to develop a reduction for BLS signatures in the multi-user setting.

In 2002, Galbraith, Malone-Lee, and Smart claimed that single-user unforgeability (EUF-CMA) tightly implies multi-user unforgeability (MU-EUF-CMA) for any Schnorr-like signature scheme [10]. However, the GMLS reduction, which explains how to construct a single-user forger given a multi-user forger, contains an error, pointed out by Bernstein in October 2015 [5]. The error arises from the single-user forger $\mathscr{F}_1$ using its challenge public key y to create each of the public keys it gives the multi-user forger $\mathscr{F}_n$. To answer each of $\mathscr{F}_n$'s signature queries, $\mathscr{F}_1$ must always query its own signing oracle for y with the same message. The analysis of $\mathscr{F}_1$'s success probability overlooks the possibility that $\mathscr{F}_n$'s forgery is for a message with which it previously queried the signing oracle (for *any* of the users). In addition to pointing out the error, Bernstein proved that single-user security for Schnorr signatures (EUF-CMA) tightly implies multi-user security for *key-prefixed* Schnorr signatures (MU-EUF-CMA) in the standard model. He also argued that such a reduction for Schnorr signatures without key-prefixing is unlikely to exist.

In November 2015, Kiltz, Masny, and Pan gave a reduction showing that *strong* single-user security (SEUF-CMA) tightly implies *strong* multi-user security (MU-SEUF-CMA) for Schnorr signatures in the random oracle model [15]. In response, Bernstein pointed out that the assumption of strong unforgeability is less well-understood: "Having to assume 'strong' unforgeability isn't as good as assuming standard unforgeability—there could be huge differences in security between these two attack targets" [4]. This work by Bernstein, Kiltz, Masny, and Pan was in the context of IETF standardization of elliptic-curve based signature schemes. Their results played a role in the Crypto Forum Research Group (CFRG)'s selection of a proposal [4], illustrating the importance of appropriate security models and highlighting the difficulty of interpreting security reductions when implementing schemes.

Kiltz, Masny, and Pan later generalized their work to reduce the security of signature schemes obtained from identification schemes via the Fiat-Shamir transform in the multi-user setting to the security of their underlying identification schemes [16]. The tightness of this security reduction is independent of the number of users and does not require key-prefixing. Other recent work proved that a tightness loss in the number of users is unavoidable for some signature schemes when reducing security in the multi-user setting to the single-user setting [2]. This result applies to adversaries who can corrupt (i.e., learn the private key of) all but one of the users.

2 BLS signatures

The security reduction for BLS signatures in the single-user setting loses tightness by a factor of q_s, the number of signature queries a forger can make. We restate the result here, and note that the tightness loss of q_s is optimal in the sense that there exists no tighter reduction [17].

[BLS security reduction [7, 8] *If solving the co-CDH problem in $G_1 \times G_2$ is (t', ϵ')-hard, then the BLS signature scheme is (t, ϵ, q_h, q_s)-secure against (single-user) existential forgery under adaptive chosen-message attacks, for*

$$\begin{aligned} t &= t' - (q_h + q_s)t_e - q_h t_m, \textit{ and} \\ \epsilon &= \epsilon' e(q_s + 1). \end{aligned}$$

For the multi-user security of any signature scheme, it is known that there exists a non-tight reduction to single-user security, where the tightness loss is the number of users [10]. A natural question is whether a tighter reduction exists for BLS. Although we were not able to find a tight reduction in the standard model without key-prefixing, our reduction—illustrated in Fig. 3—loses tightness by a factor of about $\min\{n, q_s + 1\}$. We omit the details of this non-tight reduction and instead supplement Fig. 3 with some intuition and an overview of the proof.

The single-user forger $\mathscr{F}_1$ embeds its challenge public key into a fraction α of the n public keys it provides to the multi-user forger $\mathscr{F}_n$. When $\mathscr{F}_n$ queries the signing oracle for a signature by a user whose public key does not depend on $\mathscr{F}_1$'s challenge public key, $\mathscr{F}_1$ can simply compute the signature. Otherwise, it answers $\mathscr{F}_n$'s signature query by forwarding it to its own signing oracle and multiplying the result by the appropriate power of the message's hash.

Although $\mathscr{F}_1$ can always answer $\mathscr{F}_n$'s signature queries, the reduction can fail even if $\mathscr{F}_n$ successfully forges a signature. For $\mathscr{F}_1$ to succeed as well, two more conditions must be met: user i^* must have a pk-dependent key, and $\mathscr{F}_n$ must not have requested a signature

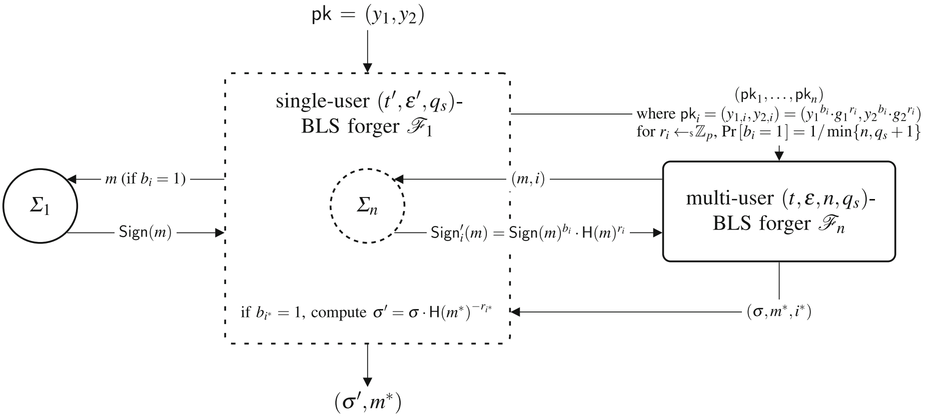

$\mathsf{pk} = (y_1, y_2)$
single-user (t', ε', q_s)-
BLS forger $\mathscr{F}_1$
$(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$
where $\mathsf{pk}_i = (y_{1,i}, y_{2,i}) = ({y_1}^{b_i} \cdot {g_1}^{r_i}, {y_2}^{b_i} \cdot {g_2}^{r_i})$
for $r_i \leftarrow_{\$} \mathbb{Z}_p$, $\Pr[b_i = 1] = 1/\min\{n, q_s + 1\}$
Σ_1
m (if $b_i = 1$)
$\mathsf{Sign}(m)$
Σ_n
(m, i)
$\mathsf{Sign}'_i(m) = \mathsf{Sign}(m)^{b_i} \cdot \mathsf{H}(m)^{r_i}$
multi-user (t, ε, n, q_s)-
BLS forger $\mathscr{F}_n$
if $b_{i^*} = 1$, compute $\sigma' = \sigma \cdot \mathsf{H}(m^*)^{-r_{i^*}}$
(σ, m^*, i^*)
(σ', m^*)

Fig. 3 Reduction from multi-user security to single-user security for BLS

on m^* by any user with a pk-dependent key. Given that $\mathscr{F}_n$ succeeded, the first condition is met with probability at least α. Given $\mathscr{F}_n$'s success and the first condition being met, the second condition is met with probability at least $(1-\alpha)^{\min\{n-1,q_s\}}$, since the number of signatures $\mathscr{F}_n$ can request on m^* is limited by the number of users and the total number of signatures it can request. Therefore, given $\mathscr{F}_n$'s success, $\mathscr{F}_1$ succeeds with probability at least $\alpha \cdot (1-\alpha)^{\min\{n-1,q_s\}}$, which is maximized for $\alpha = 1/\min\{n, q_s+1\}$ and corresponds to a tightness gap of about $e \cdot \min\{n, q_s+1\}$.

This non-tight reduction leaves open the question of whether there is an attack on multi-user BLS that is much faster than on single-user BLS. Recall that for Schnorr signatures, there were two approaches to making a multi-user to single-user security reduction tight: strong unforgeability and prefixing keys to messages. Since BLS signatures are not probabilistic, we try the second approach: we establish a reduction from multi-user key-prefixed BLS security to single-user BLS security. Let BLS-KP refer to the key-prefixed variant of BLS, whose details are in Table 2. We obtain a result for BLS analogous to Bernstein's for Schnorr signatures. See Fig. 4 for an illustration of this tight reduction and Theorem 1 for its details.

Theorem 1 (**Reduction from multi-user BLS-KP security to single-user BLS security**) *If the BLS signature scheme is resistant to (t', ϵ, q_s)-existential forgery under adaptive chosen-message attacks in the single-user setting (EUF-CMA), then BLS-KP is resistant to (t, ϵ, n, q_s)-existential forgery under adaptive chosen-message attacks in the multi-user setting (MU-EUF-CMA), for any t, n,and q_s satisfying*

$$t = t' + (2n + q_s + 1)(t_m + t_e).$$

Table 2 Summary of the BLS-KP signature scheme

Setup, KeyGen	Same as BLS (Table 1)
Sign(sk, m)	$\sigma = \mathsf{H}(\mathsf{pk}\|\|m)^{\mathsf{sk}} \in G_1$
Ver(pk, σ, m)	$e(\mathsf{H}(\mathsf{pk}\|\|m), y_2) \overset{?}{=} e(\sigma, g_2)$

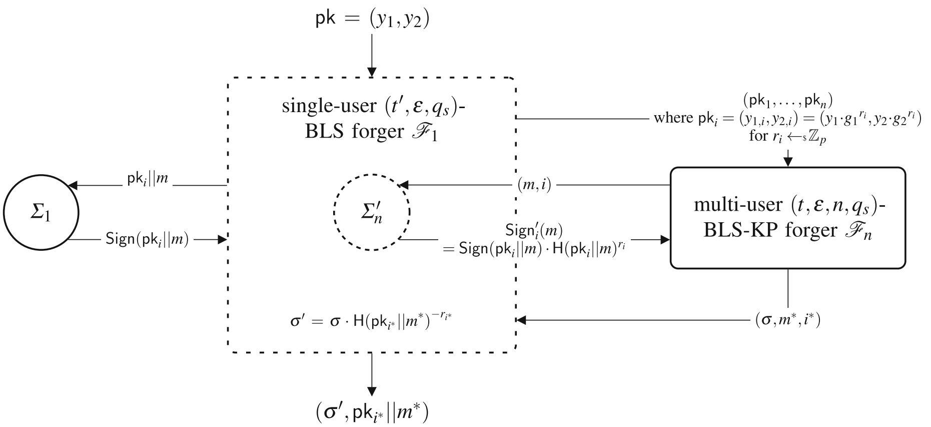

$\mathsf{pk} = (y_1, y_2)$
single-user (t', ε, q_s)-BLS forger $\mathscr{F}_1$
$(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$
where $\mathsf{pk}_i = (y_{1,i}, y_{2,i}) = (y_1 \cdot g_1{}^{r_i}, y_2 \cdot g_2{}^{r_i})$
for $r_i \leftarrow_\$ \mathbb{Z}_p$
$\mathsf{pk}_i||m$
Σ_1
$\mathsf{Sign}(\mathsf{pk}_i||m)$
Σ'_n
(m, i)
$\mathsf{Sign}'_i(m)$
$= \mathsf{Sign}(\mathsf{pk}_i||m) \cdot \mathsf{H}(\mathsf{pk}_i||m)^{r_i}$
multi-user (t, ε, n, q_s)-BLS-KP forger $\mathscr{F}_n$
$\sigma' = \sigma \cdot \mathsf{H}(\mathsf{pk}_{i^*}||m^*)^{-r_{i^*}}$
(σ, m^*, i^*)
$(\sigma', \mathsf{pk}_{i^*}||m^*)$

Fig. 4 Reduction from multi-user BLS-KP security to single-user BLS security

Proof We proceed in the usual manner, by proving the contrapositive: given a multi-user (t, ϵ, n, q_s)-forger $\mathscr{F}_n$ for BLS-KP, we build a single-user (t', ϵ, q_s)-forger $\mathscr{F}_1$ for BLS. $\mathscr{F}_1$ receives a challenge public key $\mathsf{pk} = (y_1, y_2)$ and has access to a signing oracle Σ_1 for pk. $\mathscr{F}_1$ gives $\mathscr{F}_n$ the following n public keys:

$$\mathsf{pk}_i = (y_{1,i}, y_{2,i}) = (y_1 \cdot g_1{}^{r_i}, y_2 \cdot g_2{}^{r_i})$$

where $r_i \leftarrow_\$ \mathbb{Z}_p$. $\mathscr{F}_1$ records (i, r_i) for each of the n public keys.

$\mathscr{F}_1$ must simulate responses from a signing oracle for $\mathscr{F}_n$, which can make q_s signature queries. To answer the query (m, i) for a signature on m by the user with key pk_i, $\mathscr{F}_1$ requests a signature on the message $\mathsf{pk}_i||m$ from Σ_1 and then computes $\mathsf{Sign}'_i(m) = \mathsf{Sign}(\mathsf{pk}_i||m) \cdot \mathsf{H}(\mathsf{pk}_i||m)^{r_i}$. This signature is valid:

$$\begin{aligned} e(\mathsf{Sign}(\mathsf{pk}_i||m) \cdot \mathsf{H}(\mathsf{pk}_i||m)^{r_i}, g_2) &= e(\mathsf{Sign}(\mathsf{pk}_i||m), g_2) \cdot e(\mathsf{H}(\mathsf{pk}_i||m), g_2{}^{r_i}) \\ &= e(\mathsf{H}(\mathsf{pk}_i||m), y_2) \cdot e(\mathsf{H}(\mathsf{pk}_i||m), g_2{}^{r_i}) \\ &= e(\mathsf{H}(\mathsf{pk}_i||m), y_{2,i}). \end{aligned}$$

After time at most t and with probability at least ϵ, $\mathscr{F}_n$ outputs a forgery (σ, m^*, i^*) that is new and valid: (m^*, i^*) was not queried to Σ_n and $\mathsf{Ver}(\mathsf{pk}_{i^*}, \sigma, m^*) = 1$, specifically, $e(\sigma, g_2) = e(\mathsf{H}(\mathsf{pk}_{i^*}||m^*), y_{2,i^*})$. Then, $\mathscr{F}_1$ computes $\sigma' = \sigma \cdot \mathsf{H}(\mathsf{pk}_{i^*}||m^*)^{-r_{i^*}}$ and outputs the forgery $(\sigma', \mathsf{pk}_{i^*}||m^*)$. This signature is a valid forgery on $\mathsf{pk}_{i^*}||m^*$ by the user with public key pk since

$$\begin{aligned} e(\sigma', g_2) &= e(\sigma, g_2) \cdot e(\mathsf{H}(\mathsf{pk}_{i^*}||m^*)^{-r_{i^*}}, g_2) \\ &= e(\mathsf{H}(\mathsf{pk}_{i^*}||m^*), y_{2,i^*}) \cdot e(\mathsf{H}(\mathsf{pk}_{i^*}||m^*), g_2{}^{-r_{i^*}}) \\ &= e(\mathsf{H}(\mathsf{pk}_{i^*}||m^*), y_2 \cdot g_2{}^{r_{i^*}} \cdot g_2{}^{-r_{i^*}}) \\ &= e(\mathsf{H}(\mathsf{pk}_{i^*}||m^*), y_2). \end{aligned}$$

There is a one-to-one correspondence between the signing queries of $\mathscr{F}_1$ and $\mathscr{F}_n$, so $\mathscr{F}_1$ made exactly q_s signing queries and never queried Σ_1 with $\mathsf{pk}_{i^*}||m^*$ since $\mathscr{F}_n$ never queried Σ'_n with (m^*, i^*). $\mathscr{F}_1$'s success probability is exactly $\mathscr{F}_n$'s success probability ϵ. Finally, $\mathscr{F}_1$'s only additional work was computing $2n + q_s + 1$ multiplications and $2n + q_s + 1$ exponentiations in G_1 or G_2, so $t' = t + (2n + q_s + 1)(t_m + t_e)$, giving the required bounds. □

By composing the BLS security reduction and Theorem 1, we can obtain a security reduction for BLS in the multi-user setting with key-prefixing. One might be tempted to conclude that key-prefixing is necessary to preserve BLS security in a multi-user setting, but it is not. In fact, the security of BLS in the multi-user setting can be directly reduced to the hardness of the co-CDH problem, with the same tightness loss as BLS in the single-user setting [14]. To see this, first consider the single-user BLS security reduction. The co-CDH solver embeds one part of the co-CDH instance in the public key it gives to the forger, and the other part in a particular fraction of the message hashes. It succeeds if it was able to answer all of the forger's signature queries, if the forger succeeded, and if the message on which the forger forged a signature was one of the special fraction of messages. In the multi-user BLS security reduction, the co-CDH solver can simply embed the first part of the co-CDH instance in *all* of the public keys it provides the BLS forger. The rest of the reduction (and thus, the analysis) is the same.

Now that we have a tight reduction relating the security of multi-user BLS (with key-prefixing) and single-user BLS, we turn our attention to the BGLS aggregate signature scheme.

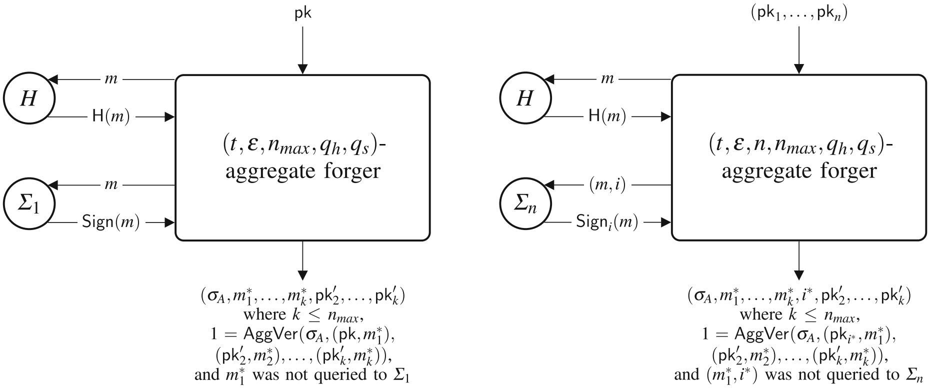
3 Aggregate signatures

Aggregate signature schemes combine multiple users' signatures on multiple (possibly different) messages. Their current chosen-key security model does not reflect the possibility that an adversary may consider it sufficient to forge a signature involving any one user, rather than a given user.

Boneh, Gentry, Lynn, and Shacham introduced aggregate signatures in 2003 [6]. These schemes allow compressing many signatures into one signature of shorter length, sometimes even independent of the number of included signatures. A general aggregate signature scheme comprises five algorithms (Setup, KeyGen, Sign, Agg, AggVer) on four sets (public keys, secret keys, messages, and signatures). Setup, KeyGen, and Sign work exactly as they do in a normal signature scheme, except that KeyGen is run once for each user. Agg takes two or more signatures and outputs one (aggregate) signature. AggVer takes a signature, a multi-set of $k \leqslant n_{max}$ public key-message pairs, and outputs 1 if the signature is a valid aggregate. While *sequential* aggregate signature schemes exist, we consider only *general* aggregate signature schemes where aggregation can be performed by anyone in any order. In the following experiments, we always assume without loss of generality that the messages in each aggregate signature are numbered so that the first one corresponds to the non-trivial component of that signature, as we describe below.

3.1 General aggregate signature security model

The original security model is the "aggregate chosen-key security model" [6], in which the adversary receives one public key generated with KeyGen and can adaptively query a signing oracle with messages of its choice. Its goal is to output a valid, non-trivial aggregate signature on $k \in [n_{max}]$ messages for which it chose $k - 1$ of the public keys. "Non-trivial" means that the first message (corresponding to the challenge public key) was not queried to the signing oracle. See Fig. 5a for a diagram of this experiment, the aggregate existential unforgeability under adaptive chosen-message attacks (A-EUF-CMA) experiment. We write pk or pk_i for public keys generated using KeyGen, and pk'_j for public keys chosen by the adversary. The keys it chooses may or may not be equal to some of the keys it received as input.


pk
H
m
H(m)
$(t, \varepsilon, n_{max}, q_h, q_s)$-
aggregate forger
Σ_1
m
Sign(m)
$(\sigma_A, m_1^*, \ldots, m_k^*, \mathsf{pk}_2', \ldots, \mathsf{pk}_k')$
where $k \le n_{max}$,
$1 = \mathsf{AggVer}(\sigma_A, (\mathsf{pk}, m_1^*),$
$(\mathsf{pk}_2', m_2^*), \ldots, (\mathsf{pk}_k', m_k^*))$,
and m_1^* was not queried to Σ_1
$(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$
H
m
H(m)
$(t, \varepsilon, n, n_{max}, q_h, q_s)$-
aggregate forger
Σ_n
(m, i)
$\mathsf{Sign}_i(m)$
$(\sigma_A, m_1^*, \ldots, m_k^*, i^*, \mathsf{pk}_2', \ldots, \mathsf{pk}_k')$
where $k \le n_{max}$,
$1 = \mathsf{AggVer}(\sigma_A, (\mathsf{pk}_{i^*}, m_1^*),$
$(\mathsf{pk}_2', m_2^*), \ldots, (\mathsf{pk}_k', m_k^*))$,
and (m_1^*, i^*) was not queried to Σ_n

(a) A-EUF-CMA experiment

(b) MU-A-EUF-CMA experiment

Fig. 5 Two types of aggregate existential forgery experiments in the random oracle model. The public keys pk and $\mathsf{pk}_1, \ldots, \mathsf{pk}_n$ are generated with KeyGen

We believe the chosen-key security model is analogous to the single-user setting for (non-aggregate) digital signatures: the adversary is given *one* public key to target. We propose a new security model that is truly a multi-user model—the forger receives n challenge public keys generated by KeyGen and it can choose which one or ones to target, eventually forging an aggregate signature on at most n_{max} messages. A valid, non-trivial aggregate signature in this model must include at least one message signed by a user with one of the challenge public keys, and the forger must not have requested a signature on this message by this user from the signing oracle. The other public keys, if any, may be challenge keys or they may have been generated by the forger. We call this model "multi-user aggregate existential forgery under adaptive chosen-message attacks" (MU-A-EUF-CMA). (See Fig. 5b.)

Chosen-key aggregate forgery is at least as hard as multi-user aggregate forgery: an aggregate forger in the chosen-key model can easily be translated to an aggregate forger in the multi-user model. For the converse, however, we do not know of a tight reduction, only one that loses tightness by a factor of the number of users, n. This straightforward, general result does not require the random oracle model.

Proposition 1 (**Reduction from multi-user to chosen-key aggregate signature security**) *If an aggregate signature scheme is resistant to aggregate* $(t, \epsilon, n_{max}, q_s)$*-existential forgery under adaptive chosen-message attacks (A-EUF-CMA), then it is resistant to multi-user aggregate* $(t, \epsilon, n, n_{max}, q_s)$*-existential forgery under adaptive chosen-message attacks (MU-A-EUF-CMA) for any* t, ϵ, n, n_{max}, *and* q_s *satisfying*

$$\begin{aligned} t &= t' - q_s t_{\mathsf{Sign}} - (n-1) t_{\mathsf{KeyGen}}, \text{ and} \\ \epsilon &= \epsilon' n \end{aligned}$$

where t_{Sign} *and* t_{KeyGen} *are the times required to run* Sign *and* KeyGen.

This result applies to any general aggregate signature scheme, and could be composed with a security reduction in the A-EUF-CMA model to obtain a security reduction in the MU-A-EUF-CMA model. For BGLS, however, it is possible to obtain a tighter security reduction by first reducing security to the underlying signature scheme, BLS, in the multi-user model.

Our reductions involving BGLS aggregate forgers are in the random oracle model. We make the following simplifying assumptions:

- When a forger requests a signature on a message from a signing oracle, it has already obtained the hash of this message from the hashing oracle.
- A forger never makes the same query twice.
- When a forger outputs a signature on a message (or messages), every message was previously hashed.

3.2 BGLS and key-prefixing

Recall that we used key-prefixing to obtain a tight reduction from multi-user BLS-KP security to single-user BLS security, as did Bernstein for Schnorr signatures [5]. Key-prefixing is also relevant to BGLS signatures, but for a different reason: to prevent a rogue-key attack. In such an attack, a malicious user claims its public key is some function of an honest user's public key, but without actually knowing the associated secret key. Table 3 summarizes the basic BGLS scheme, which must be modified or restricted to prevent a rogue-key attack.

The rogue key attack, identified in the original BGLS paper [6], works as follows. Suppose honest user 1 has public key $\mathsf{pk}_1 = (y_{1,1}, y_{2,1})$. Malicious user 2 can pick any integer $x \in \mathbb{Z}_p$ and publish $\mathsf{pk}_2 = ({g_1}^x \cdot {y_{1,1}}^{-1}, {g_2}^x \cdot {y_{2,1}}^{-1})$ as its public key. Then, user 2 can compute $\sigma_A = \mathsf{H}(m)^x$ for any message m and claim that it is an aggregate signature on m comprising signatures by both itself and honest user 1. This signature is valid since

$$\begin{aligned} e\left(\mathsf{H}(m), y_{2,1}\right) \cdot e\left(\mathsf{H}(m), y_{2,2}\right) &= e\left(\mathsf{H}(m), y_{2,1} \cdot {g_2}^x \cdot {y_{2,1}}^{-1}\right) \\ &= e\left(\mathsf{H}(m), {g_2}^x\right) \\ &= e\left(\sigma_A, g_2\right). \end{aligned}$$

This attack applies to the basic BGLS scheme as defined in Table 3 in both the original A-EUF-CMA model and our new MU-A-EUF-CMA model. Boneh, Gentry, Lynn, and Shacham suggested applying one of the following three countermeasures:

- Require users to prove knowledge of their private keys (e.g., by disclosing their private keys to a trusted party).

Table 3 Basic BGLS aggregate signature scheme [6]

Setup	$G_1 = \langle g_1 \rangle$, $G_2 = \langle g_2 \rangle$, G_T, prime order p
	$\mathsf{H} : \{0,1\}^* \to G_1$, full-domain
	$e : G_1 \times G_2 \to G_T$, bilinear
KeyGen	$x \leftarrow_\$ \mathbb{Z}_p$
	$\mathsf{sk} = x \in \mathbb{Z}_p$
	$\mathsf{pk} = (y_1, y_2) = ({g_1}^x, {g_2}^x) \in G_1 \times G_2$
$\mathsf{Sign}(\mathsf{sk}, m)$	$\sigma = \mathsf{H}(m)^{\mathsf{sk}} \in G_1$
$\mathsf{Agg}(\sigma_1, \ldots, \sigma_k)$	$\sigma_A = \prod_{i=1}^k \sigma_i \in G_1$
$\mathsf{AggVer}(\sigma_A, (\mathsf{pk}_1, m_1), \ldots, (\mathsf{pk}_k, m_k))$	$\prod_{i=1}^k e(\mathsf{H}(m_i), y_{2,i}) \overset{?}{=} e(\sigma_A, g_2)$

- Require users to prove possession of their private keys (e.g., by signing random messages that will never be used in practice).
- Require all of the messages in one aggregate signature to be distinct.

The authors suggested that the last option might be the simplest, and further suggested that to achieve distinctness of messages, a user could simply prefix its public key to a message, creating an "enhanced" or "key-prefixed" message, before hashing it. Then, the distinctness requirement would apply only to *each user's* messages in the aggregate, rather than *all* messages in the aggregate.

In their 2007 paper, Bellare, Namprempre, and Neven pointed out that while hashing enhanced messages eliminates the rogue-key attack, the requirement for distinct enhanced messages is restrictive and unnecessary [3]. There may be applications where multiple signatures by the same user on the same message need to be aggregated. They suggested that an "unrestricted" scheme—with no requirement for enhanced messages to be distinct—is more practical and is sufficient for preventing the rogue-key attack. See Table 4 for this unrestricted, key-prefixed variant "BGLS-KP."

Bellare, Namprempre, and Neven presented a tight reduction from the unforgeability of BGLS-KP to the unforgeability of BLS in the random oracle model [3]. Composing this reduction with the standard BLS security reduction yields a security reduction for BGLS-KP that loses tightness by a factor of q_s. Then, using a technique of Katz and Wang [13], they presented a tight security reduction for a variant of BGLS with key-prefixing, "BGLS-KP-KW," where each signer further enhances a message before hashing and signing it by also prefixing a random bit of its choice (Table 5).

In the next section, we determine whether similar results hold for BGLS in our multi-user security model. First, we examine whether there is also a tight reduction from BGLS-KP security in the multi-user model to BLS-KP security in the multi-user model. Next, we determine whether the Katz-Wang trick is enough to yield a tight security reduction for BGLS-KP-KW in the multi-user model.

3.3 BGLS security in a truly multi-user setting

In the standard model, it is not obvious how to reduce the security of multi-user BGLS-KP to the security of multi-user BLS-KP: a BLS forger would need to isolate one component of the BGLS forger's aggregate signature, which requires being able to compute signatures on messages by users whose keys are chosen by the BGLS-KP forger. In the random oracle model, however, it is possible to obtain a tight reduction from BGLS-KP to BLS-KP security in the multi-user setting, as the next theorem proves. See Fig. 6 for an illustration of the reduction.

Theorem 2 (**Reduction from multi-user BGLS-KP security to multi-user BLS-KP security**) *If the BLS-KP signature scheme is resistant to multi-user* $(t', \epsilon, n, q_h, q_s')$*-existential forgery under adaptive chosen-message attacks (MU-EUF-CMA), then the BGLS-KP aggregate signature scheme is resistant to multi-user* $(t, \epsilon, n, n_{max}, q_h, q_s)$*-existential*

Table 4 BGLS-KP aggregate signature scheme [3]

Setup, KeyGen, Agg	Same as BGLS (Table 3)
Sign(sk, m)	Same as BLS-KP (Table 2)
AggVer(σ_A, (pk$_1$, m_1), ..., (pk$_k$, m_k))	$\prod_{i=1}^{k} e(\mathsf{H}(\mathsf{pk}_i \| m_i), y_{2,1}) \overset{?}{=} e(\sigma_A, g_2)$

Table 5 BGLS-KP-KW aggregate signature scheme [3]

Setup, KeyGen	Same as BGLS (Table 3)
$\mathsf{Sign}(\mathsf{sk}, m)$	$(\sigma = \mathsf{H}(b\|\|\mathsf{pk}\|\|m)^{\mathsf{sk}}, b) \in G_1 \times \{0, 1\}$
$\mathsf{Agg}((\sigma_1, b_1), \ldots, (\sigma_n, b_k))$	$(\sigma_A = \prod_{i=1}^{k} \sigma_i, b_1, \ldots, b_k) \in G_1 \times \{0, 1\}^k$
$\mathsf{AggVer}(\sigma_A, (\mathsf{pk}_1, m_1), \ldots, (\mathsf{pk}_k, m_k), b_1, \ldots, b_k)$	$\prod_{i=1}^{k} e(\mathsf{H}(b_i\|\|\mathsf{pk}_i\|\|m_i), y_{2,i}) \overset{?}{=} e(\sigma_A, g_2)$

aggregate forgery under adaptive chosen-message attacks (MU-A-EUF-CMA), for any $t, \epsilon, n, n_{max}, q_h$,and q_s satisfying

$$t = t' - (q_h + n_{max} + 1)t_e + (n_{max} - 1)t_m \textit{ and}$$
$$q_s = q_s' - n_{max} + 1.$$

Proof We proceed by proving the contrapositive: given a multi-user BGLS-KP $(t, \epsilon, n, n_{max}, q_h, q_s)$-aggregate forger $\mathscr{F}_A$, we build a multi-user $(t', \epsilon, n, q_h, q_s')$-forger $\mathscr{F}$ for BLS-KP. $\mathscr{F}$ receives n challenge public keys $(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$ and can query a hashing oracle H and a signing oracle Σ_n. $\mathscr{F}$ gives $\mathscr{F}_A$ the same n public keys and must simulate a hashing oracle H' and signing oracle Σ_n'. When $\mathscr{F}_A$ makes a hash query, the reply depends on the message's format:

$$\mathsf{H}'(m) = \begin{cases} \mathsf{H}(m') & \text{if } m = \mathsf{pk}||m' \text{ for some } \mathsf{pk} \in \{\mathsf{pk}_1, \ldots, \mathsf{pk}_n\} \\ {g_1}^r, r \leftarrow_{\$} \mathbb{Z}_p & \text{else.} \end{cases}$$

In the first case, $\mathscr{F}$ must query H. In the second case, $\mathscr{F}$ records (m, r). When $\mathscr{F}_A$ queries Σ_n' with (m, i), $\mathscr{F}$ in turn queries Σ_n with (m, i) and replies to $\mathscr{F}_A$ with $\mathsf{Sign}_i'(m) = \mathsf{Sign}_i(m)$.

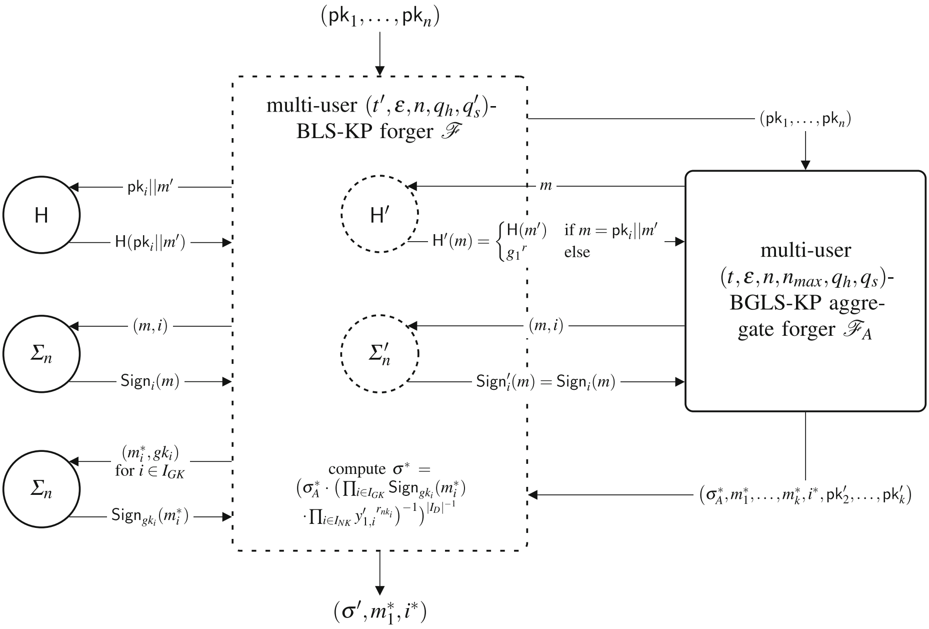

$(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$
multi-user $(t', \varepsilon, n, q_h, q_s')$-BLS-KP forger $\mathscr{F}$
$(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$
H
$\mathsf{pk}_i||m'$
$\mathsf{H}(\mathsf{pk}_i||m')$
H′
m
$\mathsf{H}'(m) = \begin{cases} \mathsf{H}(m') & \text{if } m = \mathsf{pk}_i||m' \\ {g_1}^r & \text{else} \end{cases}$
multi-user $(t, \varepsilon, n, n_{max}, q_h, q_s)$-BGLS-KP aggregate forger $\mathscr{F}_A$
Σ_n
(m, i)
$\mathsf{Sign}_i(m)$
Σ_n'
(m, i)
$\mathsf{Sign}_i'(m) = \mathsf{Sign}_i(m)$
Σ_n
(m_i^*, gk_i) for $i \in I_{GK}$
$\mathsf{Sign}_{gk_i}(m_i^*)$
compute $\sigma^* = (\sigma_A^* \cdot (\prod_{i \in I_{GK}} \mathsf{Sign}_{gk_i}(m_i^*) \cdot \prod_{i \in I_{NK}} {y'_{1,i}}^{r_{nk_i}})^{-1})^{|I_D|^{-1}}$
$(\sigma_A^*, m_1^*, \ldots, m_k^*, i^*, \mathsf{pk}_2', \ldots, \mathsf{pk}_k')$
(σ', m_1^*, i^*)

Fig. 6 Reduction from multi-user BGLS-KP security to multi-user BLS-KP security

After time at most t and with probability at least ϵ, $\mathscr{F}_A$ outputs a valid aggregate forgery $(\sigma_A^*, m_1^*, \dots, m_k^*, i^*, \mathsf{pk}_2', \dots, \mathsf{pk}_k')$ for some $k \in [n_{max}]$, where (m_1^*, i^*) was not queried to Σ_n'. For simplicity, let $\mathsf{pk}_1' = \mathsf{pk}_{i^*}$. Partition the indices $[k]$ into the following three sets of "duplicates," "new keys," and "given keys":

- $I_D := \{i \in [k] : m_i^* = m_1^* \text{ and } \mathsf{pk}_i' = \mathsf{pk}_1'\}$
- $I_{NK} := \{i \in [k] : \mathsf{pk}_i' \notin \{\mathsf{pk}_1, \dots, \mathsf{pk}_n\}\}$
- $I_{GK} := \{i \in [k] \setminus I_D : \mathsf{pk}_i' = \mathsf{pk}_{gk_i} \text{ for some } gk_i \in [n]\}$

The set I_D contains at least one element, 1. For each i in I_{NK}, $\mathscr{F}$ looks up the logarithm of $\mathsf{H}'(\mathsf{pk}_i'||m_i^*)$ to base g_1, i.e., the value of r_{nk_i} such that $\mathsf{H}'(\mathsf{pk}_i'||m_i^*) = {g_1}^{r_{nk_i}}$. For each i in I_{GK}, $\mathscr{F}$ queries the signing oracle Σ_n with (m_i^*, gk_i) to get $\mathsf{Sign}_{gk_i}(m_i^*)$. Since $\mathscr{F}_A$ output a valid forgery, σ_A^* satisfies the following equations:

$$\begin{aligned}
&e(\sigma_A^*, g_2)\\
&= \prod_{i \in I_D} e(\mathsf{H}(\mathsf{pk}_{i^*}||m_1^*), y_{2,i^*}) \cdot \prod_{i \in I_{NK}} e({g_1}^{r_{nk_i}}, y_{2,i}') \cdot \prod_{i \in I_{GK}} e(\mathsf{H}(\mathsf{pk}_{gk_i}||m_i^*), y_{2,gk_i})\\
&= e(\mathsf{H}(\mathsf{pk}_{i^*}||m_1^*), y_{2,i^*})^{|I_D|} \cdot \prod_{i \in I_{NK}} e({y_{1,i}'}^{r_{nk_i}}, g_2) \cdot \prod_{i \in I_{GK}} e(\mathsf{Sign}_{gk_i}(m_i^*), g_2)\\
&= e(\mathsf{H}(\mathsf{pk}_{i^*}||m_1^*)^{|I_D|}, y_{2,i^*}) \cdot e\left(\prod_{i \in I_{NK}} {y_{1,i}'}^{r_{nk_i}} \cdot \prod_{i \in I_{GK}} \mathsf{Sign}_{gk_i}(m_i^*), g_2\right).
\end{aligned}$$

Therefore, $\mathscr{F}$ is able to compute the following signature on (m_1^*, i^*):

$$\sigma^* = \left(\sigma_A^* \cdot \left(\prod_{i \in I_{NK}} {y_{1,i}'}^{r_{nk_i}} \cdot \prod_{i \in I_{GK}} \mathsf{Sign}_{gk_i}(m_i^*)\right)^{-1}\right)^{|I_D|^{-1} \mod p}.$$

$|I_D|^{-1} \mod p$ exists as long as $|I_D| < p$, which is a reasonable assumption since otherwise $\mathscr{F}$ could find a secret key by trial exponentiation. $\mathscr{F}$ outputs (σ^*, m_1^*, i^*), which is valid because $e(\sigma^*, g_2) = e(\mathsf{H}(\mathsf{pk}_{i^*}||m_1^*), y_{2,i^*}))$.

The additional work done by $\mathscr{F}$ was computing at most $q_h + n_{max} + 1$ exponentiations and $n_{max} - 1$ multiplications in G_1, so $t' \leqslant t + (q_h + n_{max} + 1)t_e + (n_{max} - 1)t_m$. $\mathscr{F}$ made at most as many hashing queries as $\mathscr{F}_A$, q_h, and it made $q_s' \leqslant q_s + n_{max} - 1$ signing queries. It always succeeds whenever $\mathscr{F}_A$ succeeds, so $\epsilon' = \epsilon$. Thus, we have built a multi-user forger for BLS-KP given a multi-user aggregate forger for BGLS-KP with the required time and query bounds. □

The previous reduction is tight. We can compose it with the reduction from multi-user BLS-KP security to single-user BLS security (Theorem 1) and the security reduction for single-user BLS security. The result is a security reduction for BGLS-KP based on the co-CDH problem that has a tightness gap of about q_s.

Given that there is a tight security reduction for BGLS in the chosen-key model due to Katz and Wang, it is natural to wonder whether this result can be lifted to the multi-user model. Our last theorem provides an affirmative answer for this variant of BGLS-KP: there is a tight security reduction, illustrated in Fig. 7, for BGLS-KP-KW in the multi-user setting.

Theorem 3 (**Security reduction for multi-user BGLS-KP-KW**) *If the co-CDH problem is (t', ϵ')-hard in $G_1 \times G_2$, then the BGLS-KP-KW aggregate signature scheme is*

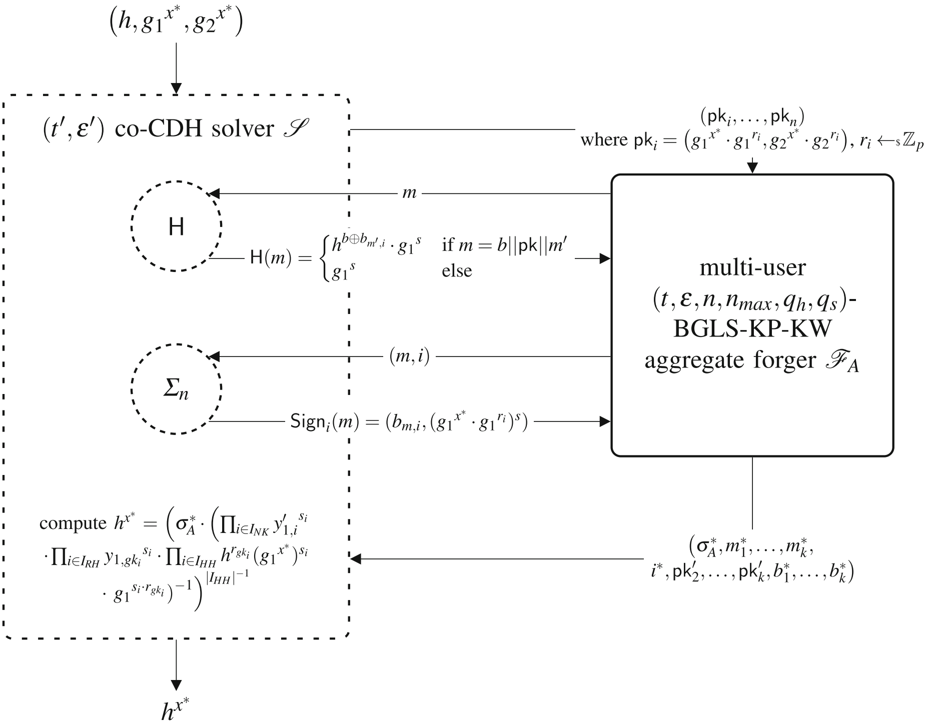

$\left(h, {g_1}^{x^*}, {g_2}^{x^*}\right)$
(t', ε') co-CDH solver $\mathscr{S}$
$(\mathsf{pk}_i, \ldots, \mathsf{pk}_n)$
where $\mathsf{pk}_i = \left({g_1}^{x^*} \cdot {g_1}^{r_i}, {g_2}^{x^*} \cdot {g_2}^{r_i}\right)$, $r_i \leftarrow_\$ \mathbb{Z}_p$
m
H
$\mathsf{H}(m) = \begin{cases} h^{b \oplus b_{m',i}} \cdot {g_1}^s & \text{if } m = b\|\mathsf{pk}\|m' \\ {g_1}^s & \text{else} \end{cases}$
multi-user
$(t, \varepsilon, n, n_{max}, q_h, q_s)$-
BGLS-KP-KW
aggregate forger $\mathscr{F}_A$
(m, i)
Σ_n
$\mathsf{Sign}_i(m) = (b_{m,i}, ({g_1}^{x^*} \cdot {g_1}^{r_i})^s)$
compute $h^{x^*} = \left(\sigma_A^* \cdot \left(\prod_{i \in I_{NK}} {y'_{1,i}}^{s_i} \cdot \prod_{i \in I_{RH}} {y_{1,gk_i}}^{s_i} \cdot \prod_{i \in I_{HH}} h^{r_{gk_i}} ({g_1}^{x^*})^{s_i} \cdot {g_1}^{s_i \cdot r_{gk_i}}\right)^{-1}\right)^{|I_{HH}|^{-1}}$
$(\sigma_A^*, m_1^*, \ldots, m_k^*,$
$i^*, \mathsf{pk}'_2, \ldots, \mathsf{pk}'_k, b_1^*, \ldots, b_k^*)$
h^{x^*}

Fig. 7 Security reduction for multi-user BGLS-KP-KW

resistant to multi-user $(t, \epsilon, n, n_{max}, q_h, q_s)$-existential aggregate forgery under adaptive chosen-message attacks (MU-A-EUF-CMA), for any $t, \epsilon, n, n_{max}, q_h$, and q_s satisfying

$$\begin{aligned} t &= t' - (2n + q_h + q_s + 2n_{max} + 1)t_e + (2n + q_h + q_s + 2n)t_m \textit{ and} \\ \epsilon &= 2\epsilon'. \end{aligned}$$

Proof We show how to build a solver $\mathscr{S}$ for the co-CDH problem given an aggregate forger $\mathscr{F}_A$ for BGLS-KP-KW. $\mathscr{S}$ receives an instance of the co-CDH problem, a triple $(h, {g_1}^{x^*}, {g_2}^{x^*}) \in G_1 \times G_1 \times G_2$, for some unknown integer $x^* \in \mathbb{Z}_p$. It must compute $h^{x^*} \in G_1$. First, $\mathscr{S}$ gives $\mathscr{F}_A$ n public keys of the form $\mathsf{pk}_i = ({g_1}^{x^*} \cdot {g_1}^{r_i}, {g_2}^{x^*} \cdot {g_2}^{r_i})$, where $r_i \leftarrow_\$ \mathbb{Z}_p$. For each of these i, $\mathscr{S}$ records (i, r_i). When $\mathscr{F}_A$ requests the hash of a message m, $\mathscr{S}$'s reply depends on the message's format:

$$\mathsf{H}(m) = \begin{cases} h^{b \bigoplus b_{m',i}} \cdot {g_1}^s & \text{if } m = b\|\mathsf{pk}\|m' \text{ where } b \in \{0, 1\}, \mathsf{pk} = \mathsf{pk}_i \text{ for an } i \in [n] \\ {g_1}^s & \text{else} \end{cases}$$

for some $s \leftarrow_\$ \mathbb{Z}_p$, where $b_{m',i} \leftarrow_\$ \{0, 1\}$. In the first case, $\mathscr{S}$ stores $(m', i, b_{m',i})$ and (m', i, b, s); in the second case, $\mathscr{S}$ stores (m, s).

When $\mathscr{F}_A$ queries the signing oracle Σ_n with (m, i), $\mathscr{S}$ always chooses to sign with $b = b_{m,i}$. It looks up the value of s corresponding to (m, i, b) and returns $\mathsf{Sign}_i(m) = (b, ({g_1}^{x^*} \cdot {g_1}^{r_i})^s)$, which is a valid signature since $({g_1}^{x^*} \cdot {g_1}^{r_i})^s = ({g_1}^s)^{x^*+r_i} = \mathsf{H}(b\|\mathsf{pk}_i\|n)^{x^*+r_i}$.

After time at most t and with probability at least ϵ, $\mathscr{F}_A$ outputs a valid aggregate forgery $(\sigma_A^*, m_1^*, \ldots, m_k^*,\ i^*, \mathsf{pk}'_2, \ldots, \mathsf{pk}'_k, b_1^*, \ldots, b_k^*)$ for some $k \in [n_{max}]$, where Σ_n never answered a query for (m_1^*, i^*) with $b = b_1^*$. For ease of notation, let $\mathsf{pk}'_1 = \mathsf{pk}_{i^*}$. Partition the indices $[k]$ into the following three sets corresponding to "new keys," "random hashes," and "h-dependent hashes":

- $I_{NK} := \{i \in [k] : \mathsf{pk}'_i \notin \{\mathsf{pk}_1, \ldots, \mathsf{pk}_n\}\}$
- $I_{RH} := \{i \in [k] \setminus I_{NK} : \mathsf{pk}'_i = \mathsf{pk}_{gk_i}$ for some $gk_i \in [n]$ and $b_i^* = b_{m_i^*,i}\}$
- $I_{HH} := \{i \in [k] \setminus I_{NK} : \mathsf{pk}'_i = \mathsf{pk}_{gk_i}$ for some $gk_i \in [n]$ and $b_i^* \neq b_{m_i^*,i}\}$

With probability $1/2$, $b_1^* \neq b_{m_1^*,i^*}$ and therefore $1 \in I_{HH}$. (If not, then $\mathscr{S}$ aborts.) For each $i \in I_{NK} \cup I_{RH}$, $\mathscr{S}$ can look up the value of s_i such that $\mathsf{H}'(b_i^*||\mathsf{pk}'_i||m_i^*) = {g_1}^{s_i}$. Similarly, for each $i \in I_{HH}$, $\mathscr{S}$ can look up the value of s_i such that $\mathsf{H}'(b_i^*||\mathsf{pk}'_i||m_i^*) = h{g_1}^{s_i}$. Since $\mathscr{F}_A$ output a valid forgery, σ_A^* satisfies the following equations:

$$
\begin{aligned}
& e(\sigma_A^*, g_2) \\
&= \prod_{i \in I_{NK}} e({g_1}^{s_i}, y'_{2,i}) \cdot \prod_{i \in I_{RH}} e({g_1}^{s_i}, y_{2,gk_i}) \cdot \prod_{i \in I_{HH}} e(h \cdot {g_1}^{s_i}, y_{2,gk_i}) \cdot \\
&= e\left(\prod_{i \in I_{NK}} {y'_{1,i}}^{s_i} \cdot \prod_{i \in I_{RH}} {y_{1,gk_i}}^{s_i} \cdot \prod_{i \in I_{HH}} (h \cdot {g_1}^{s_i})^{\mathsf{sk}_{gk_i}}, g_2 \right) \\
&= e\left(\prod_{i \in I_{NK}} {y'_{1,i}}^{s_i} \cdot \prod_{i \in I_{RH}} {y_{1,gk_i}}^{s_i} \cdot \prod_{i \in I_{HH}} (h \cdot {g_1}^{s_i})^{x^*+r_{gk_i}}, g_2 \right) \\
&= e\left(h^{x^*|I_{HH}|} \cdot \prod_{i \in I_{NK}} {y'_{1,i}}^{s_i} \cdot \prod_{i \in I_{RH}} {y_{1,gk_i}}^{s_i} \cdot \prod_{i \in I_{HH}} h^{r_{gk_i}} ({g_1}^{x^*})^{s_i} {g_1}^{s_i r_{gk_i}}, g_2 \right).
\end{aligned}
$$

$|I_{HH}|^{-1} \mod p$ exists if $|I_{HH}| < p$, a reasonable assumption, in which case the co-CDH solver $\mathscr{S}$ can compute the desired value:

$$
h^{x^*} = \left(\sigma_A^* \left(\prod_{i \in I_{NK}} {y'_{1,i}}^{s_i} \prod_{i \in I_{RH}} {y_{1,gk_i}}^{s_i} \prod_{i \in I_{HH}} h^{r_{gk_i}} ({g_1}^{x^*})^{s_i} {g_1}^{s_i r_{gk_i}} \right)^{-1} \right)^{|I_{HH}|^{-1}} .
$$

$\mathscr{S}$ had to compute $2n + q_h + q_s + 2n_{max} + 1$ exponentiations and $2n + q_h + q_s + 2n$ multiplications in G_1 or G_2. $\mathscr{S}$ succeeds whenever $\mathscr{F}_A$ does and $b_1^* \neq b_{m_1^*,i^*}$, so $\epsilon' \geq \epsilon/2$, as required. □

Although the length of a BGLS-KP-KW aggregate signature increases by 1 bit for each component, Bellare, Namprempre, and Neven argue that the BGLS-KP-KW scheme could be as efficient as BGLS-KP: the tighter reduction means that a smaller prime p can be used for the same level of security [3].

4 Conclusions

Inspired by the recent analysis of Schnorr signatures in the multi-user setting, we examined reductions for BLS and BGLS signatures in the multi-user setting. We obtained a tight reduction from the multi-user security of BLS-KP, a key-prefixed variant of BLS, to BLS in the single-user setting. We introduced a notion of security (MU-A-EUF-CMA) for general aggregate signature schemes in the multi-user setting, which we believe is more realistic than the current standard. The security of any general aggregate signature scheme in this new multi-user setting reduces to its security in the original chosen-key setting with a tightness loss of the number of users. For BGLS, it is possible to do better: we presented a tight reduction from multi-user BGLS-KP security to multi-user BLS-KP security in the random-oracle model. BGLS-KP is not only a natural extension of BLS-KP, which has a tight reduction to single-user BLS, but BGLS-KP also avoids a known rogue-key attack and has no requirement for the distinctness of (enhanced) messages. Composing this reduction with our first result—the tight multi-user BLS-KP to single-user BLS reduction—and with the standard BLS security reduction yields a security reduction for BGLS-KP with a tightness gap of q_s. Finally, we presented a tight security reduction for BGLS-KP-KW, the Katz-Wang variant of BGLS with key-prefixing, in the multi-user setting. The tightness gap of this reduction is only 2, but it is in the random oracle model. It would be interesting to perform a similar analysis on the security models for sequential aggregate signature schemes.

Although the importance of developing appropriate security models may be well known, interpreting the tightness of security reductions is still difficult. In this paper, we proved that prefixing a random bit to each enhanced message makes the BGLS-KP security reduction tight in the random oracle model. However, without these single bits, which are sent in the clear with the signature, the security reduction may lose tightness by a factor of up to q_s, the number of signature queries an adversary can make—which could be as much as 2^{20}. The question of which of these two results should guide the choice of parameter sizes in practice is difficult to answer.

Acknowledgements This work is supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No. H2020-MSCA-ITN-2014-643161 ECRYPT-NET. The author thanks Kenny Paterson for his comments on an earlier version of this work, Alfred Menezes for his guidance on related work completed in the Department of Combinatorics and Optimization at the University of Waterloo as part of the MMath programme, and the reviewers for their valuable comments.

References

1. An, J.H., Dodis, Y., Rabin, T.: On the security of joint signature and encryption. In: Knudsen, L.R. (ed.) Advances in Cryptology – EUROCRYPT 2002, Lecture Notes in Computer Science, vol. 2332, pp. 83–107. Springer, Heidelberg (2002)
2. Bader, C., Jager, T., Li, Y., Schäge, S.: On the impossibility of tight cryptographic reductions. In: Fischlin, M., Coron, J.S. (eds.) Advances in Cryptology – EUROCRYPT 2016: 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II, pp. 273–304. Springer, Berlin (2016). https://doi.org/10.1007/978-3-662-49896-5_10
3. Bellare, M., Namprempre, C., Neven, G.: Unrestricted aggregate signatures. In: Arge, L., Cachin, C., Jurdzinski, T., Tarlecki, A. (eds.) ICALP 2007: 34th International Colloquium on Automata, Languages and Programming, Lecture Notes in Computer Science, vol. 4596, pp. 411–422. Springer, Heidelberg (2007)

4. Bernstein, D.J.: [Cfrg] key as message prefix =⟩ multi-key security cfrg@ietf.org mailing list (2015). https://www.ietf.org/mail-archive/web/cfrg/current/msg07628.html
5. Bernstein, D.J.: Multi-user Schnorr security, revisited. Cryptology ePrint Archive Report 2015/996 (2015). http://eprint.iacr.org/2015/996
6. Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and verifiably encrypted signatures from bilinear maps. In: Biham, E. (ed.) Advances in Cryptology – EUROCRYPT 2003, Lecture Notes in Computer Science, vol. 2656, pp. 416–432. Springer, Heidelberg (2003)
7. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the weil pairing. In: Boyd, C. (ed.) Advances in Cryptology – ASIACRYPT 2001, Lecture Notes in Computer Science, vol. 2248, pp. 514–532. Springer, Heidelberg (2001)
8. Chatterjee, S., Hankerson, D., Knapp, E., Menezes, A.: Comparing two pairing-based aggregate signature schemes. Des. Codes Crypt. **55**(2), 141–167 (2009). https://doi.org/10.1007/s10623-009-9334-7
9. Chatterjee, S., Menezes, A., Sarkar, P.: Another look at tightness. In: Miri, A., Vaudenay, S. (eds.) SAC 2011: 18th Annual International Workshop on Selected Areas in Cryptography, Lecture Notes in Computer Science, vol. 7118, pp. 293–319. Springer, Heidelberg (2012)
10. Galbraith, S., Malone-Lee, J., Smart, N.P.: Public key signatures in the multi-user setting. Inf. Process. Lett. **83**(5), 263–266 (2002). https://doi.org/10.1016/S0020-0190(01)00338-6
11. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. Comput. **17**(2), 281–308 (1988)
12. Heninger, N., Durumeric, Z., Wustrow, E., Halderman, J.A.: Mining your Ps and Qs: Detection of widespread weak keys in network devices. In: Proceedings of the 21st USENIX Security Symposium (2012)
13. Katz, J., Wang, N.: Efficiency improvements for signature schemes with tight security reductions. In: Jajodia, S., Atluri, V., Jaeger, T. (eds.) ACM CCS 03: 10th Conference on Computer and Communications Security, pp. 155–164. ACM Press, New York (2003)
14. Kiltz, E.: Private communication (2016)
15. Kiltz, E., Masny, D., Pan, J.: Schnorr signatures in the multi-user setting. Cryptology ePrint Archive Report 2015/1122 (2015). http://eprint.iacr.org/2015/1122
16. Kiltz, E., Masny, D., Pan, J.: Optimal Security Proofs for Signatures from Identification Schemes. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology – CRYPTO 2016: 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II, pp. 33–61. Springer Berlin Heidelberg, Berlin (2016)
17. Knapp, E.: On Pairing-Based Signature and Aggregate Signature Schemes. University of Waterloo, MMath thesis (2008)
18. Menezes, A., Smart, N.: Security of signature schemes in a multi-user setting. Des. Codes Crypt. **33**(3), 261–274 (2004). https://doi.org/10.1023/B:DESI.0000036250.18062.3f
19. Schnorr, C.P.: Efficient Identification and Signatures for Smart Cards. In: Brassard, G. (ed.) Advances in Cryptology – CRYPTO'89, Lecture Notes in Computer Science, vol. 435, pp. 239–252. Springer, Heidelberg (1990)