Security of BLS and BGLS Signatures in a Multi-User Setting

Marie-Sarah Lacharité

Royal Holloway, University of London
marie-sarah.lacharite.2015@rhul.ac.uk

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. Using 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. 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, which 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 in the standard model. 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 co-CDH problem. All of our results for BLS and BGLS use type III pairings.

Keywords: security models, signatures, aggregate signatures, multi-user setting

1 Introduction

It is important to have security models that reflect real-world conditions. While single-user settings are much easier to analyse, they exclude many real-world attacks. For example, it is possible to factor RSA keys by computing their GCDs if two moduli share a prime factor. 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 [8].

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 the single-user setting was not precisely known [3, 4, 12].

In this paper, we examine the security reduction for Boneh-Lynn-Shacham (BLS, [6]) signatures in the multi-user model. We were not able to find a tight

reduction from multi-user BLS security to single-user BLS security in the standard model; our reduction is non-tight by a factor of about $\max\{n, q_s + 1\}$, about the number of users or the maximum number of signature queries allowed. However, we do obtain a tight reduction in the standard model from multi-user *key-prefixed* BLS security to single-user BLS security.

Next, we examine the Boneh-Gentry-Lynn-Shacham (BGLS) aggregate signature scheme, which is a natural extension of BLS and was the first proposed aggregate signature scheme [5]. We introduce a truly multi-user security model for general aggregate signature schemes, as opposed to the chosen-key model of BGLS. We present a tight reduction from key-prefixed multi-user BGLS security to key-prefixed multi-user BLS security in the standard model. Finally, 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 [11]. Figure 1 summarizes our results.

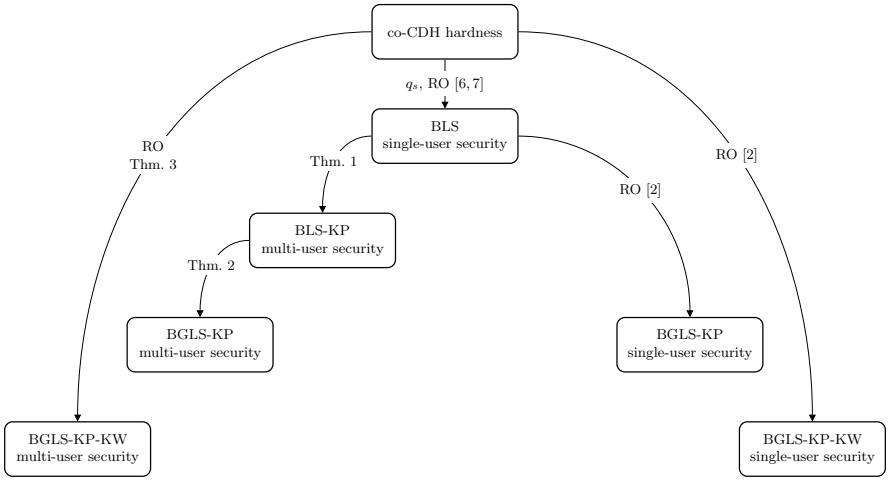


Fig. 1: Overview of our results for BLS and BGLS and how they compare to reductions in the single-user setting. All reductions are tight other than co-CDH hardness → single-user BLS security. All reductions are in the standard model unless otherwise indicated with "RO" for "random oracle."

1.1 Notation, Assumptions, and Review of Schnorr and BLS

See Table 1 for an overview of the Schnorr [14] and BLS [6] signature schemes. For simplicity, we omit the role of the security parameter in Setup.

Although the BLS scheme was introduced for symmetric pairings, where $G_1 = G_2$, we use the modified scheme due to Chatterjee et al. that works for asymmetric pairings where no efficiently computable isomorphism from G_2 to

Table 1: Summary of the Schnorr and BLS signature schemes

	Schnorr [14]	**BLS [6]**
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	$x \leftarrow_\$ \mathbb{Z}_p$ $\mathsf{sk} = x \in \mathbb{Z}_p$ $\mathsf{pk} = g^x \in G$	$x \leftarrow_\$ \mathbb{Z}_p$ $\mathsf{sk} = x \in \mathbb{Z}_p$ $\mathsf{pk} = (y_1, y_2) = ({g_1}^x, {g_2}^x) \in G_1 \times G_2$
$\mathsf{Sign}(\mathsf{sk}, m)$	$k \leftarrow_\$ \mathbb{Z}_p$ $\sigma = (h, s)$ $= (\mathsf{H}(g^k \| m), \mathsf{sk} \cdot h + k)$ $\in \mathbb{Z}_p \times \mathbb{Z}_p$	$\sigma = \mathsf{H}(m)^{\mathsf{sk}} \in G_1$
$\mathsf{Ver}(\mathsf{pk}, \sigma, m)$	$h \stackrel{?}{=} \mathsf{H}(g^s \cdot \mathsf{pk}^{-h} \| m)$	$e(\mathsf{H}(m), y_2) \stackrel{?}{=} e(\sigma, g_2)$

G_1 is known ("type III" pairings) [7]. The security of the BLS signature scheme in the type III setting depends on the hardness of solving the **computational co-Diffie-Hellman (co-CDH) problem** in $G_1 \times G_2$: given ${g_2}^x \in G_2$, ${g_1}^x \in G_1$, and $h \in G_1$, compute $h^x \in G_1$. We say that a problem is $(\mathbf{t}, \epsilon)$**-hard** if there exists no adversary who can solve it in time at most t with probability at least ϵ. It is not obvious from the definition of this scheme why the public key must contain both ${g_1}^x$ and ${g_2}^x$, since only the latter is used for verification. The first part of the public key is necessary in the reduction from BLS forgery to solving the co-CDH problem (the opposite direction of that which we examine in this paper).

We denote by t_m and t_e 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 denote by $x \| y$ the concatenation of (the binary representations of) x and y.

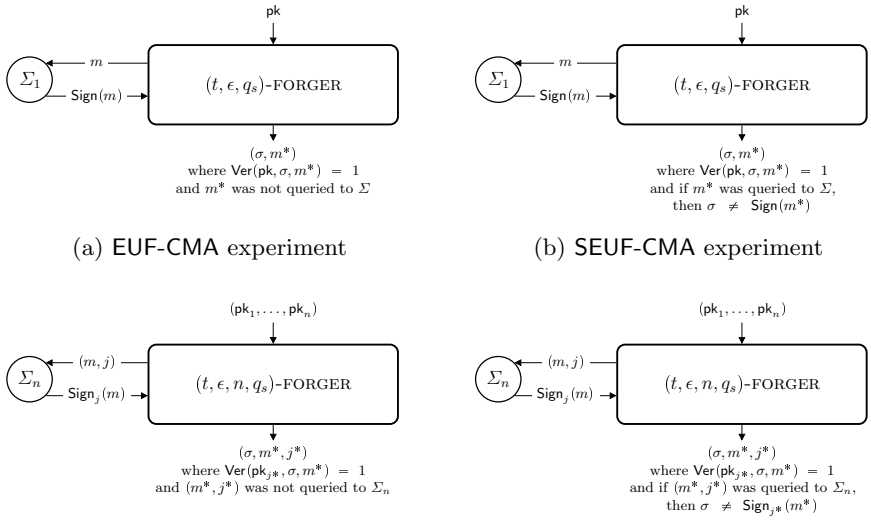
1.2 Standard and Strong Unforgeability

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 [10]. A digital signature scheme is **(single-user)** $(\mathbf{t}, \epsilon, \mathbf{q_s})$**-existentially unforgeable under adaptive chosen-message attacks** ($\mathsf{EUF\text{-}CMA}$) if there exists no forger $\mathcal{F}$ 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 a signature on a message it did not submit to the signing oracle with probability at least ϵ. See Figure 2a for a diagram of the $\mathsf{EUF\text{-}CMA}$ experiment.

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 $\mathcal{F}$, 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 Figure 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.



(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. Similar experiments exist in the random oracle model, where the forgers can also make at most q_h queries to a hashing oracle.

1.3 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.

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 $\mathcal{F}$ 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 by one of the n users on a message for which it did not submit to the signing oracle for this user. See Figure 2c.

Again, for probabilistic signature schemes, there is 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 $\mathcal{F}$, 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 Figure 2d.

1.4 Multi-User Schnorr Signature Security: a Brief History

In this section, we review the security of Schnorr signatures in the multi-user model. We later 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 [9]. However, the GMLS reduction, which explains how to construct a single-user forger given a multi-user forger, contains an error that Bernstein pointed out in October 2015 [4].

The error arises from the single-user forger $\mathcal{F}_1$ using its challenge public key y to create each of the public keys it gives the multi-user forger $\mathcal{F}_n$. To answer each of $\mathcal{F}_n$'s signature queries, $\mathcal{F}_1$ must always query its own signing oracle for y with the same message. The analysis of $\mathcal{F}_1$'s success probability overlooks the possibility that $\mathcal{F}_n$'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 [12]. However, as Bernstein pointed out, "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" [3].

This recent work by Bernstein and Kiltz, Masny, and Pan was in the context of IETF standardization of elliptic-curve based signature schemes. These results played a role in the Crypto Forum Research Group (CFRG)'s selection of a proposal [3], illustrating the importance of appropriate security models and highlighting the difficulty of interpreting security reductions when implementing schemes.

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.

Theorem 1 (BLS security reduction [6, 7]). *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*

$$t = t' - (q_h + q_s)t_e - q_h t_m, \text{ and}$$
$$\epsilon = \epsilon' e(q_s + 1).$$

The tightness loss of q_s is optimal in the sense that there exists no tighter reduction [13].

Next, we examine whether it is possible to obtain a tight reduction from the security of BLS in the multi-user setting to its security the single-user setting. For Schnorr signatures, it is possible with either strong unforgeability in the random oracle model or key-prefixing in the standard model. Our first attempt for BLS is in the standard model without key-prefixing and yields the reduction illustrated in Figure 3. This reduction is not tight, so we omit its details.

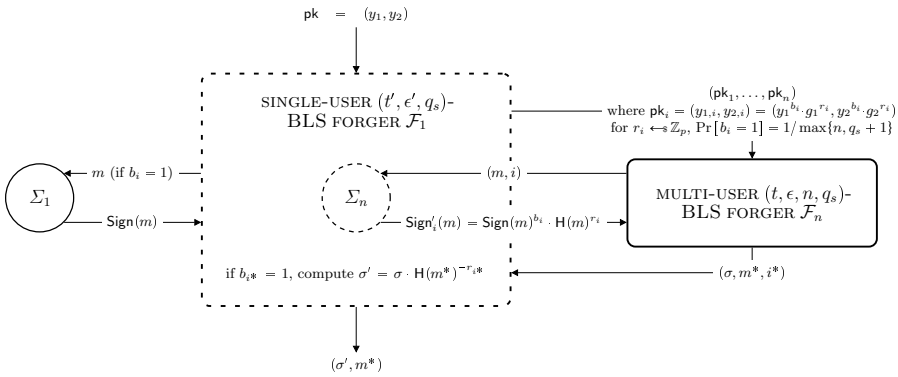


Fig. 3: Reduction from multi-user security to single-user security for BLS. For the single-user forger to succeed, the multi-user forger must not have requested a signature on m^* by any user with a pk-dependent key and user i^* must have a pk-dependent key. The tightness gap is about $e \cdot \max\{n, q_s + 1\}$.

Unlike the reduction for strong unforgeability of Schnorr signatures, this reduction is not tight. Such a non-tight reduction could mean that there is an attack on multi-user BLS that is faster than on single-user BLS, or simply that we have not found a tighter reduction yet. However, comparing the BLS reduction

to the Schnorr reduction, the second possibility seems unlikely. The principal reason that the BLS reduction is not as tight as the Schnorr reduction seems to be that BLS signatures are deterministic while the Schnorr reduction is for strong unforgeability. If BLS signatures were probabilistic, then there would at least exist the possibility that $\mathcal{F}_1$ succeeds even if $\mathcal{F}_n$ requests signatures on m^* by users whose public keys depend on pk. A second, minor difference is that no forgeries produced by $\mathcal{F}_n$ could allow $\mathcal{F}_1$ to extract the secret signing key sk. In the Schnorr reduction, extraction of the secret key x is possible because one part of each signature is a linear function of x.

Our second reduction uses a key-prefixed variant of BLS that we call BLS-KP. See Table 2 for the details of BLS-KP. We obtain a result analogous to Bernstein's for Schnorr signatures. See Figure 4 for an illustration of this tight reduction and Theorem 2 for its details.

Table 2: Summary of the BLS-KP signature scheme

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

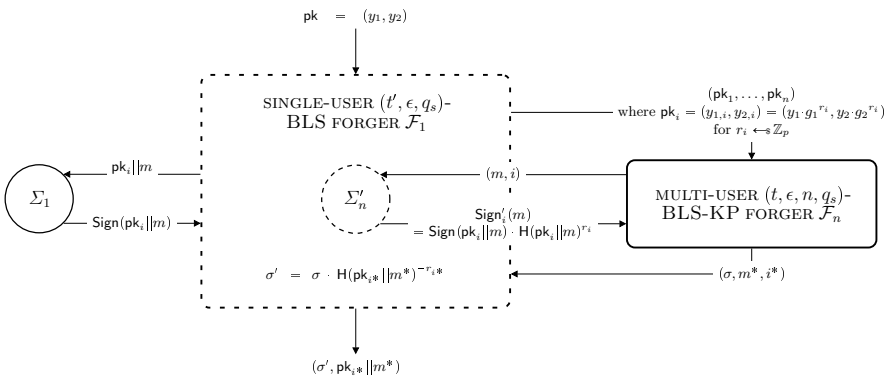


Fig. 4: Reduction from multi-user BLS-KP security to single-user BLS security.

Theorem 2 (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

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

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

$\mathcal{F}_1$ must simulate responses from a signing oracle for $\mathcal{F}_n$, who can make q_s signature queries. To answer the query (m, i) for a signature on m by the user with key pk_i, $\mathcal{F}_1$ 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, $\mathcal{F}_n$ 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, $\mathcal{F}_1$ 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 $\mathcal{F}_1$ and $\mathcal{F}_n$, so $\mathcal{F}_1$ made exactly q_s signing queries and never queried Σ_1 with $\mathsf{pk}_{i^*}||m^*$ since $\mathcal{F}_n$ never queried Σ'_n with (m^*, i^*). $\mathcal{F}_1$'s success probability is exactly $\mathcal{F}_n$'s success probability ϵ. Finally, $\mathcal{F}_1$'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. □

Now that we have a tight reduction relating the security of multi-user BLS-KP to single-user BLS, we examine the BGLS aggregate signature scheme.

3 Aggregate Signatures

The multi-user setting is natural for aggregate signature schemes, which combine multiple users' signatures on multiple (possibly different) messages. However, we think the current chosen-key security model may not reflect this.

Boneh, Gentry, Lynn, and Shacham introduced aggregate signatures in 2003 [5]. Aggregate signature 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, but KeyGen is run once for each user. Agg takes 2 or more signatures and outputs 1 (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.

There also exist *sequential* aggregate signature schemes, but we consider only *general* aggregate signature schemes where aggregation can be performed by anyone in any order. Table 3 summarizes the BGLS scheme.

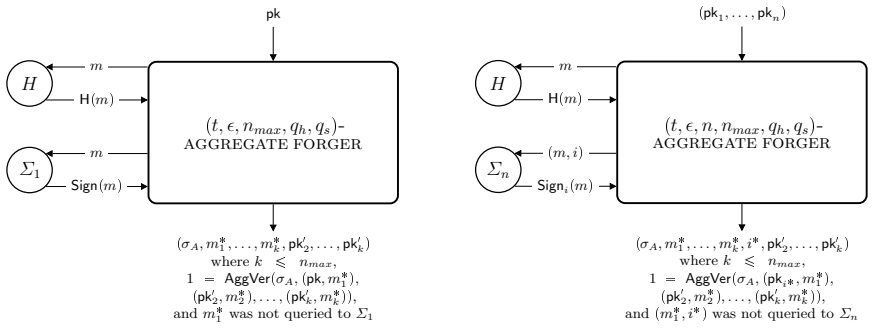
Table 3: BGLS aggregate signature scheme [5]

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	$x \leftarrow\!\!\$\, \mathbb{Z}_p$ $\mathsf{sk} = x \in \mathbb{Z}_p$ $\mathsf{pk} = (y_1, y_2) = ({g_1}^x, {g_2}^x) \in G_1 \times G_2$
$\mathsf{Sign}(\mathsf{sk}, m)$	$\sigma = \mathsf{H}(m)^{\mathsf{sk}} \in G_1$
$\mathsf{Agg}(\sigma_1, \ldots, \sigma_k)$	$\sigma_A = \prod_{i=1}^{k} \sigma_i \in G_1$
$\mathsf{AggVer}(\sigma_A, (\mathsf{pk}_1, m_1), \ldots, (\mathsf{pk}_k, m_k))$	$\prod_{i=1}^{k} e(\mathsf{H}(m_i), y_{2,i}) \stackrel{?}{=} e(\sigma_A, g_2)$

3.1 General Aggregate Signature Security Model

The original security model is the "aggregate chosen-key security model" [5]. The adversary receives one public key and can adaptively query a signing oracle with messages of its choice. Its goal is to output a valid, non-trivial aggregate signature, $k \in [n_{max}]$ messages, and $k-1$ public keys of its choice. "Non-trivial" means that the first message (corresponding to the challenge public key) was not queried to the signing oracle. See Figure 5a for a diagram of this experiment, the aggregate existential unforgeability under adaptive chosen-message attacks (A-EUF-CMA) experiment.

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 keys and can choose which one(s) to target, eventually forging an aggregate signature on at most n_{max} messages. We call this model



(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.

"multi-user aggregate existential forgery under adaptive chosen-message attacks" (MU-A-EUF-CMA). See Figure 5b.

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 Key-Prefixing and the Rogue-Key Attack

We used key-prefixing to get a tight reduction from multi-user BLS-KP security to single-user BLS security, as did Bernstein for Schnorr signatures [4]. Key-prefixing is also relevant to BGLS signatures, but for a different reason: to prevent a rogue-key attack, where a malicious user claims its public key is some function of an honest user's public key, without actually knowing the associated secret key. BGLS as defined in Table 3 is vulnerable to the following rogue-key attack, identified in the original BGLS paper [5]. 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

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

Boneh, Gentry, Lynn, and Shacham suggested the following three countermeasures:

- Require users to prove knowledge of their private keys (e.g., by disclosing their private keys to a trusted party).
- 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 suggest that the last option might be the simplest, and further suggest that to achieve distinctness of messages, a user could simply prepend its public key to a message, creating an "enhanced" or "key-prefixed" message, before hashing it. Then, the distinctness requirement applies only to *each user's* messages in the aggregate, rather than *all* messages in the aggregate.

In their 2007 paper, Bellare, Namprempre, and Neven point out that while hashing enhanced messages eliminates the rogue-key attack, the requirement for distinct enhanced messages is restrictive and unnecessary [2]. There may be applications where multiple signatures by the same user on the same message need to be aggregated. Bellare, Namprempre, and Neven suggest 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."

Table 4: BGLS-KP aggregate signature scheme [2]

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

Bellare, Namprempre, and Neven present a tight reduction from the unforgeability of BGLS-KP to the unforgeability of BLS in the random oracle model. 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 [11], they present a tight security reduction for a variant of BGLS with key-prefixing, "BGLS-KP-KW," where each signer further enhances a message before signing it by also prepending a random bit of its choice.

In the next section, we determine whether similar results hold for BGLS in a truly 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.

Table 5: BGLS-KP-KW aggregate signature scheme [2]

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

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 Figure 6 for an illustration of the reduction.

Theorem 3 (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 aggregate forgery under adaptive chosen-message attacks (MU-A-EUF-CMA), for any $t, \epsilon, n, n_{max}, q_h,$ and q_s satisfying*

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

Proof. We prove the contrapositive: given a multi-user BGLS-KP $(t, \epsilon, n, n_{max}, q_h, q_s)$-aggregate forger $\mathcal{F}_A$, we build a multi-user $(t', \epsilon, n, q_h, q'_s)$-forger $\mathcal{F}$ for BLS-KP. $\mathcal{F}$ 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. $\mathcal{F}$ gives $\mathcal{F}_A$ the same n public keys and must simulate a hashing oracle H' and signing oracle Σ'_n. When $\mathcal{F}_A$ 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, $\mathcal{F}$ must query H. In the second case, $\mathcal{F}$ records (m, r). When $\mathcal{F}_A$ queries Σ'_n with (m, i), $\mathcal{F}$ in turn queries Σ_n with (m, i) and replies to $\mathcal{F}_A$ with $\mathsf{Sign}'_i(m) = \mathsf{Sign}_i(m)$.

After time at most t and with probability at least ϵ, $\mathcal{F}_A$ outputs a valid aggregate forgery $(\sigma^*_A, m^*_1, \ldots, m^*_k, i^*, \mathsf{pk}'_2, \ldots, \mathsf{pk}'_k)$ for some $k \in [n_{max}]$, where

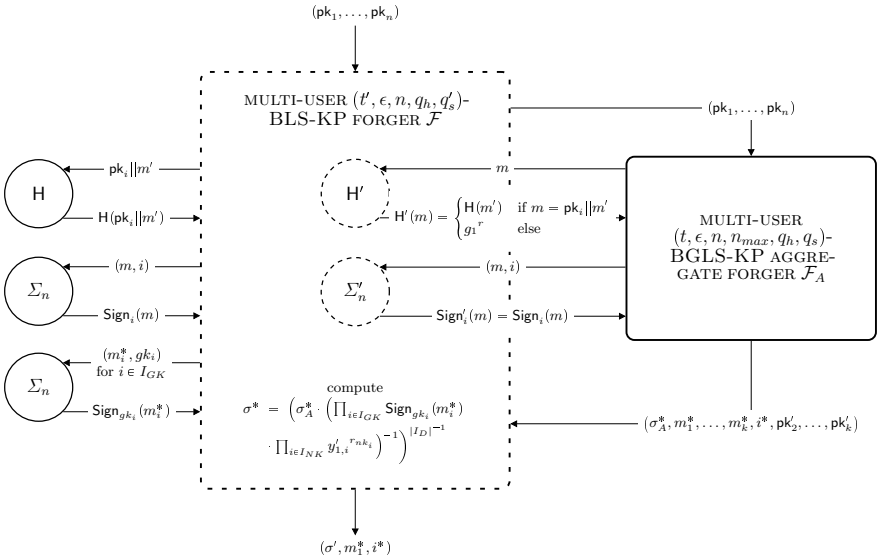


Fig. 6: Reduction from multi-user BGLS-KP security to multi-user BLS-KP security.

(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":

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

The set I_D contains at least one element, 1. For each i in I_{NK}, $\mathcal{F}$ 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}, $\mathcal{F}$ queries the signing oracle Σ_n with (m_i^*, gk_i) to get $\mathsf{Sign}_{gk_i}(m_i^*)$. Since $\mathcal{F}_A$ 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, $\mathcal{F}$ 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 $\mathcal{F}$ could find a secret key by trial exponentiation. $\mathcal{F}$ 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 $\mathcal{F}$ was computing at most $q_h + n_{max} + 1$ exponentiations and $n_{max} - 1$ multiplications in G_1, so $t' \leqslant t + (q_h + n_{max} + 1)t_e + (n_{max} - 1)t_m$. $\mathcal{F}$ made at most as many hashing queries as $\mathcal{F}_A$, q_h, and it made $q'_s \leqslant q_s + n_{max} - 1$ signing queries. It always succeeds whenever $\mathcal{F}_A$ 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 2) 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.

Is it possible to get a tighter reduction by directly reducing the security of BGLS-KP in the multi-user model to the hardness of the co-CDH problem, like it is in the chosen-key model? Our last theorem provides an affirmative answer for a slight variant of BGLS-KP: there is a tight security reduction, illustrated in Figure 7, for BGLS-KP-KW in the multi-user setting.

Theorem 4 (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 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*

$$t = t' - (2n + q_h + q_s + 2n_{max} + 1)t_e + (2n + q_h + q_s + 2n)t_m \text{ and}$$
$$\epsilon = 2\epsilon'.$$

Proof. We show how to build a solver $\mathcal{S}$ for the co-CDH problem given an aggregate forger $\mathcal{F}_A$ for BGLS-KP-KW. $\mathcal{S}$ 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, $\mathcal{S}$ gives $\mathcal{F}_A$ 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, $\mathcal{S}$ records (i, r_i). When $\mathcal{F}_A$ requests the hash of a message m, $\mathcal{S}$'s reply depends on the message's format:

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

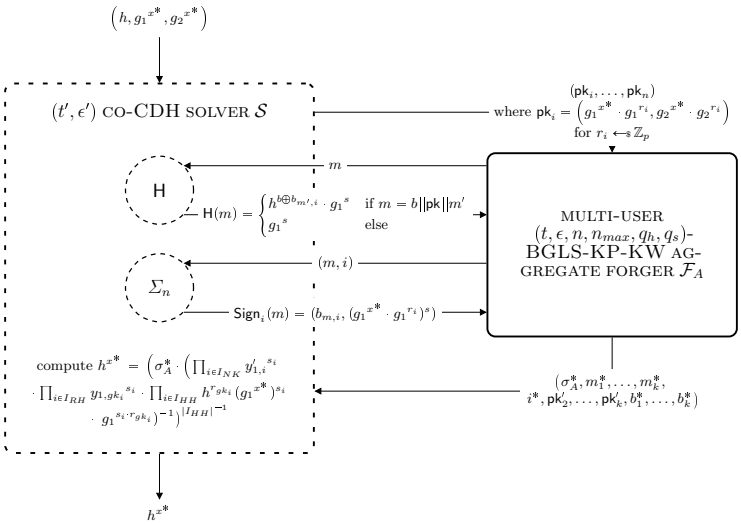


Fig. 7: Security reduction for multi-user BGLS-KP-KW.

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

When $\mathcal{F}_A$ queries the signing oracle Σ_n with (m, i), $\mathcal{S}$ 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, $\mathcal{F}_A$ 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":

- $I_{NK} := \{i \in [k] : \mathsf{pk}'_i \notin \{\mathsf{pk}_1, \ldots, \mathsf{pk}_n\}\}$
- $I_{RH} := \{i \in [k] \backslash I_{NK} : \mathsf{pk}'_i = \mathsf{pk}_{gk_i}$ for some $gk_i \in [n]$ and $b_i^* = b_{m_i^*,i}\}$
- $I_{HH} := \{i \in [k] \backslash I_{NK} : \mathsf{pk}'_i = \mathsf{pk}_{gk_i}$ 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 $\mathcal{S}$ aborts.) For each $i \in I_{NK} \cup I_{RH}$, $\mathcal{S}$ 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 $i \in I_{HH}$, $\mathcal{S}$ 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 $\mathcal{F}_A$ 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 $\mathcal{S}$ can compute the desired value:

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

$\mathcal{S}$ 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. $\mathcal{S}$ succeeds whenever $\mathcal{F}_A$ does and $b_1^* \neq b_{m_1^*, i^*}$, so $\epsilon' \geqslant \epsilon/2$, 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 [2].

4 Conclusions

Inspired by the recent analysis of Schnorr signatures in the multi-user setting, we examined reductions for BLS in the multi-user setting. We were not able to find a tight reduction from multi-user security of BLS to single-user security; the loss of tightness in our reduction was about $\max\{n, q_s + 1\}$, where the number of public keys n a forger sees may be huge in practice. Instead, 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.

Next, we introduced a notion of security (MU-A-EUF-CMA) for general aggregate signature schemes in the multi-user setting. We presented a tight reduction from multi-user BGLS-KP security to multi-user BLS security in the standard 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 of distinct (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, however, 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 prepending 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.

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 (Apr / May 2002)
2. 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 (Jul 2007)
3. Bernstein, D.J.: [cfrg] key as message prefix $\Rightarrow$ multi-key security. cfrg@ietf.org mailing list (November 2015), `https://www.ietf.org/mail-archive/web/cfrg/current/msg07628.html`
4. Bernstein, D.J.: Multi-user Schnorr security, revisited. Cryptology ePrint Archive, Report 2015/996 (2015), `http://eprint.iacr.org/`
5. 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 (May 2003)
6. 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 (Dec 2001)
7. Chatterjee, S., Hankerson, D., Knapp, E., Menezes, A.: Comparing two pairing-based aggregate signature schemes. Designs, Codes and Cryptography 55(2), 141–167 (2009), `http://dx.doi.org/10.1007/s10623-009-9334-7`

8. 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 (Aug 2012)
9. Galbraith, S., Malone-Lee, J., Smart, N.P.: Public key signatures in the multi-user setting. Inf. Process. Lett. 83(5), 263–266 (Sep 2002), http://dx.doi.org/10.1016/S0020-0190(01)00338-6
10. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal on Computing 17(2), 281–308 (Apr 1988)
11. 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 (Oct 2003)
12. Kiltz, E., Masny, D., Pan, J.: Schnorr signatures in the multi-user setting. Cryptology ePrint Archive, Report 2015/1122 (2015), http://eprint.iacr.org/
13. Knapp, E.: On Pairing-Based Signature and Aggregate Signature Schemes. MMath thesis, University of Waterloo (2008)
14. 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 (Aug 1990)