A preliminary version is merged with [37] and [29], and published as [1] for presentation in Crypto 2010.

Signing on Elements in Bilinear Groups for Modular Protocol Design

Masayuki Abe* Kristiyan Haralambiev**† Miyako Ohkubo***‡

* Information Sharing Platform Laboratories, NTT Corporation, Japan
abe.masayuki@lab.ntt.co.jp

** Computer Science Department, New York University, U.S.A.
kkh@cs.nyu.edu

*** NICT, Japan
m.ohkubo@nict.or.jp

Abstract

A signature scheme is called *structure-preserving* if its verification keys, messages, and signatures are group elements and the verification predicate is a conjunction of pairing product equations. We answer to the open problem of constructing a constant-size structure-preserving signature scheme. The security is proven in the standard model based on a novel non-interactive assumption that can be justified and has an optimal bound in the generic bilinear group model. We also present efficient structure-preserving signature schemes with advanced properties including signing unbounded number of group elements, allowing simulation in the common reference string model, signing messages from mixed groups in the asymmetric bilinear group setting, and strong unforgeability. Among many applications, we show two examples; an adaptively secure round optimal blind signature scheme and a group signature scheme with efficient concurrent join. As a bi-product, several homomorphic trapdoor commitment schemes and one-time signature schemes are presented, too. In combination with the Groth-Sahai non-interactive proof system, these schemes contribute to give efficient instantiations to modular constructions of cryptographic protocols.

Keywords: Structure-Preserving Signatures, Simulatable Signatures, Groth-Sahai Proofs, Blind Signatures

1 Introduction

Background. Cryptographic protocols often allow modular constructions that combine general building blocks such as commitments, encryption, signatures, and zero-knowledge proofs. While modular design is useful to show feasibility of cryptographic tasks and also to illustrate a comprehensible framework, efficient instantiations are sometimes left as a next challenge. Some cryptographic

†Work done while visiting NTT Information Sharing Platform Laboratories.
‡Work done while working in NTT Information Sharing Platform Laboratories.

tasks find "cleverly crafted" efficient solutions dedicated for their own purposes. Nevertheless, modular construction makes it easier and can be a good alternative for comparison when the building blocks have reasonable instantiations.

A combination of digital signatures and non-interactive zero-knowledge proofs of knowledge appears frequently in privacy-protecting cryptographic protocols such as blind signatures [28, 3], group signatures [6, 41, 8], anonymous credential systems [5], verifiably encrypted signatures [13, 49], non-interactive group encryption [26] and so on. An efficient non-interactive proof system in the standard model, however, has been absent until recently. In [39], Groth and Sahai presented the first (and currently the only) efficient non-interactive proof system based on bilinear mapping. Their proof system (GS proofs for short) exerts its full power as a proof of knowledge system when the proof statement is described as a conjunction of relations described by pairing product equations and when the witnesses consists of group elements [1]. To be compatible with the GS proof system, a signature scheme is required to provide properties such that (1) the verification keys, messages, and signatures are elements of bilinear groups, and (2) the verification predicate is a conjunction of pairing products. A signature scheme with these properties is called *structure-preserving* in [1] and have numerous applications.

Related Work. There are efficient signature schemes, e.g., [11, 22, 5, 20], whose all but one components are group elements. Research on structure-preserving signature schemes was initiated in [35]. In [35], Groth showed the first feasibility result based on the decision linear assumption (DLIN) [12]. The signature scheme yields a signature of size $\mathcal{O}(k)$ when the message consists of k group elements. While it is remarkable that the security can be shown based on a simple standard assumption, the scheme is not practical due to its large constant factor. Based on the q-Hidden LRSW assumption for asymmetric bilinear groups, Green and Hohenberger presented a structure-preserving signature scheme that provides security against random message attacks [34]. Unfortunately, an extension to the chosen message security is not known. In [29], Fuchsbauer presented a scheme based on (a variant of) the Double Hidden Strong Diffie-Hellman Assumption (DHSDH) from [30]. Their scheme is pretty efficient but has limited generality since a trusted set-up is necessary and the messages must be in a special form called a Diffie-Hellman pair. In [26], Cathalo, Libert and Yung showed a scheme based on a combination of the Hidden Strong Diffie-Hellman Assumption (HSDH), Flexible Diffie-Hellman Assumption, and the DLIN assumption. Their signature consists of $9k+4$ group elements and it is left as an open problem to construct constant-size signatures.

Our Contribution. We present the first *constant-size* structure-preserving signature scheme for messages of general bilinear group elements. A signature consists only of 7 group elements regardless of the size of the message. For a message $(m_1, \ldots, m_k)$, a signature (z, r, s, t, u, v, w) fulfills the verification equations

$$A = e(g_z, z)\, e(g_r, r)\, e(s, t) \prod_{i=1}^{k} e(g_i, m_i), \text{ and}$$

$$B = e(h_z, z)\, e(h_u, u)\, e(v, w) \prod_{i=1}^{k} e(h_i, m_i)$$

determined by the verification key.

[1] Disjunction can be handled in somewhat tricky way with extra computation and storage[19]. When the witness is a scalar, it is possible to preserve the proof of knowledge property, but it requires a bit-wise treatment and results in proofs growing linearly in the security parameter.

The unforgeability against adaptive chosen message attacks is proven in the standard model based on a novel non-interactive assumption called the Simultaneous Flexible Pairing Assumption (SFP). It is a strong, i.e., so-called "q-type" assumption like the popular Strong Diffie-Hellman Assumption (SDH) [11]. On the positive side, SFP is a rare strong assumption that achieves the optimal quadratic security bound when analyzed in the generic group model [53] while SDH and its variations suffer from a cubic bound. (We refer to [27] and [45] for a risk and discussion about non-optimality in the generic model.) Another positive point is that SFP implies the Simultaneous Double Pairing Assumption (SDP), a simple assumption implied by DLIN and that allows to build useful commitment schemes that could be smoothly integrated into constructions based on SFP. On the negative side, SFP is more complex than (H)SDH. Nevertheless, we enlighten the bright side and hope that SFP be considered as a reasonable alternative for primitive designs when only group elements are involved.

We then explore variations and a few applications as follows.

- In Section 5, based on the observation that the constant-size signatures allow unbounded "signature chaining", we present a signature scheme that signs *unbounded-size* messages. Since the message space of the resulting scheme covers the verification key space, this extension gives an automorphic signature scheme [29], which has number of interesting high-level applications coupled with GS proofs.

- In Section 6, we addresses *simulatability* in the common reference string (CRS) model. With simulatable signatures, a simulator can create signatures for arbitrary messages by using the trapdoor for the CRS. Such a property is useful in building adaptively secure protocols where a simulator has to have correct signatures without having help from a corrupted signer [3]. The resulting scheme gives an efficient instantiation to (adaptively secure variant of) Fischlin's round-optimal blind signature framework [28, 3] which we present in Section 9.1. It has been an open problem since Crypto'06 and considered as difficult [44].

- In Section 7, we present a scheme that works in the asymmetric setting where the symmetric external Diffie-Hellman (SXDH) assumption holds. This setting is of interest as the GS proof system can provide better efficiency and some protocols may demand such a setting. We stress that it is not trivial to sign a message consisting of elements from both $\mathbb{G}_1$ and $\mathbb{G}_2$ since there are no efficient mappings between both groups, and straightforward independent signing allows a forgery.

- Finally, in Section 8, we show a variation that provides strong unforgeability with constant-size signatures.

As a bi-product, we present several homomorphic trapdoor commitment schemes that are useful in coupling with our signature schemes and the GS proofs. One of them is fully GS-compatible, i.e., its commitment-key, message, commitment, and decommitment are in $\mathbb{G}_1$ and $\mathbb{G}_2$ and the verification predicate consists of pairing product equations. It is the first such scheme that binds multiple messages at once.

ORGANIZATION. After introducing necessary notations and notions in Section 2, the main constant-size signature scheme is presented in Section 4. It is followed by variations in Section 5, Section 6, and Section 7. We show outline of high-level applications in Section 9. Some of the formal proofs are shown in Appendix. The commitment schemes are presented in Appendix B.

2 Preliminaries

2.1 Common Setup with Bilinear Groups

Let $\Lambda := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, \tilde{g})$ be a description of groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ of prime order p equipped with efficient bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. It also includes a random generator g of $\mathbb{G}_1$ and $\tilde{g}$ of $\mathbb{G}_2$. By $\mathbb{G}_1^*$ we denote $\mathbb{G}_1 \setminus \{1_{\mathbb{G}_1}\}$, and the same for $\mathbb{G}_2^*$ and $\mathbb{G}_T^*$. By Λ_{sym} we denote a special case of Λ where $\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G}$. Similarly, Λ_{xdh} denotes a case where the Decision Diffie-Hellman (DDH) assumption holds for $\mathbb{G}_1$ ($\mathrm{DDH}_{\mathbb{G}_1}$ in short). This setting implies that there is no efficiently computable mapping $\mathbb{G}_1 \to \mathbb{G}_2$. And Λ_{sxdh} denotes a case where the DDH assumption holds for both $\mathbb{G}_1$ and $\mathbb{G}_2$. This means that no efficient mapping is available for either direction. The Λ_{xdh} and Λ_{sxdh} settings are usually referred to as the (Symmetric) External Diffie-Hellman Assumption [52, 12, 32, 54]. For differences of these settings in practice, we refer to [31]. A scheme (or an assumption or a proof) designed and proven in one setting may not necessarily go through in a different setting. In particular, if the scheme is for Λ_{sym} and uses the homomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$, it does not work, or not known to be secure when used with Λ_{xdh} or Λ_{sxdh}. We treat Λ as a common parameter implicitly given to all algorithms of interest. However, we present our constructions with care so that it is clear in which setting they work and are secure.

2.2 Digital Signatures

Definition 1 (Digital Signature Scheme). A digital signature scheme SIG is a set of algorithms ($\mathsf{SIG.Key}$, $\mathsf{SIG.Sign}$, $\mathsf{SIG.Vrf}$) such that:

$\mathsf{SIG.Key}(1^\lambda)$: A key generation algorithm that takes security parameter 1^λ and generates a verification key vk and a signing key sk. Message space $\mathcal{M}$ is associated to vk.

$\mathsf{SIG.Sign}(sk, m)$: A signature generation algorithm that computes a signature σ for input message m by using signing key sk.

$\mathsf{SIG.Vrf}(vk, m, \sigma)$: A verification algorithm that outputs 1 for acceptance or 0 for rejection according to the input.

A signature scheme must provide correctness in the sense that if the key pair and a signature on a message are generated legitimately, $\mathsf{SIG.Vrf}$ returns 1. In this paper, algorithms works over common bilinear setting Λ. The security parameter 1^λ allows $\mathsf{SIG.Key}$ to implicitly select Λ of appropriate size. $\mathsf{SIG.Key}$ may also take some other parameters if necessary.

We use standard notion of existential unforgeability against adaptive chosen message attacks [33] (EUF-CMA in short) formally defined as follows.

Definition 2 (Existential Unforgeability against Adaptive Chosen Message Attacks). A signature scheme is existentially unforgeable against adaptive chosen message attacks if, for any polynomial-time adversary $\mathcal{A}$, the following experiment returns 1 with negligible probability.

Experiment :

$(vk, sk) \leftarrow \mathsf{SIG.Key}(1^\lambda)$

$(m^\star, \sigma^\star) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{sign}}}(vk)$

Return 1 if $m^\star \notin Q_m$ and $1 \leftarrow \mathsf{SIG.Vrf}(vk^\star, m^\star, \sigma^\star)$. Return 0, otherwise.

$\mathcal{O}_{\mathsf{sign}}$ is the signing oracle that takes message m and returns $\sigma \leftarrow \mathsf{SIG.Sign}(sk, m)$. Q_m is the messages submitted to $\mathcal{O}_{\mathsf{sign}}$. By requiring $(m^\star, \sigma^\star) \notin Q_{m,\sigma}$ where $Q_{m,\sigma}$ is pairs of a message and a signature observed by $\mathcal{O}_{\mathsf{sign}}$, we have the notion of Strong EUF-CMA (denoted by sEUF-CMA for short).

2.3 Assumptions

We start with introducing a simple assumption, called Double Pairing Assumption (DBP), that holds in asymmetric bilinear setting, i.e., $\Lambda \in \{\Lambda_{\mathsf{xdh}}, \Lambda_{\mathsf{sxdh}}\}$.

Assumption 1 (Double Pairing Assumption (DBP)). Given $\Lambda \in \{\Lambda_{\mathsf{xdh}}, \Lambda_{\mathsf{sxdh}}\}$ and $(g_z, g_r) \leftarrow \mathbb{G}_1^{*2}$, it is hard to find $(z, r) \in \mathbb{G}_2^* \times \mathbb{G}_2^*$ such that

$$1 = e(g_z, z)\; e(g_r, r). \tag{1}$$

It is obvious that DBP does not hold for $\Lambda = \Lambda_{\mathsf{sym}}$ since $(z, r) = (g_r^{-1}, g_z) \neq (1, 1)$ fulfills the relation. On the other hand, we can show that DBP holds for $\Lambda \in \{\Lambda_{\mathsf{xdh}}, \Lambda_{\mathsf{sxdh}}\}$ where DDH is assumed hard in $\mathbb{G}_1$.

Theorem 1. *If $DDH_{\mathbb{G}_1}$ holds for Λ, then DBP holds for Λ.*

The proof is by a straightforward reduction and given in Appendix A.1.

We note that the DBP assumption could be viewed as a simpler version of the Simultaneous Triple Pairing Assumption (STP) [37]. The DBP assumption was introduced in an earlier version of this work, and independently in [38] (personal communication) by Groth, who also showed explicitly that DBP implies STP.

Next is an extension of DBP, called Simultaneous Double Pairing Assumption (SDP), which is a weaker assumption and can be justified in any setting $\Lambda \in \{\Lambda_{\mathsf{sym}}, \Lambda_{\mathsf{xdh}}, \Lambda_{\mathsf{sxdh}}\}$ by a standard argument in the generic bilinear group model.

Assumption 2 (Simultaneous Double Pairing Assumption (SDP)). Given Λ and $(g_z, h_z, g_r, h_u) \leftarrow \mathbb{G}_1^{*4}$, it is hard to find $(z, r, u) \in \mathbb{G}_2^{*3}$ such that

$$1 = e(g_z, z)\; e(g_r, r) \qquad \text{and} \qquad 1 = e(h_z, z)\; e(h_u, u). \tag{2}$$

As shown in [26], SDP is implied by DLIN.

Next we introduce a novel assumption by extending SDP so that it should be hard to find another answer given several answers. Observe that, given an answer to an instance of SDP, one can easily yield more answers by exploiting the linearity of the relation to be satisfied. We eliminate such a linearity by multiplying random pairings to both sides of the equations in (2). Intuition is that, it should be hard to merge two random pairings $e(s, t)\, e(s', t')$ into one equivalent pairing $e(s'', t'')$. We call such a random part *flexible* as random pairings can be easily randomized or combined when their relation with respect to the same bases is known.

Assumption 3 (Simultaneous Flexible Pairing Assumption (SFP)). Let Λ be a common parameter and let g_z, h_z, g_r, and h_u be random generators of $\mathbb{G}_1$. Let $(a, \tilde{a}), (b, \tilde{b})$ be random pairs in $\mathbb{G}_1 \times \mathbb{G}_2$. For $j = 1, \ldots, q$, let $R_j = (z, r, s, t, u, v, w)$ that satisfies

$$e(a, \tilde{a}) = e(g_z, z)\; e(g_r, r)\; e(s, t) \quad \text{and} \quad e(b, \tilde{b}) = e(h_z, z)\; e(h_u, u)\; e(v, w). \tag{3}$$

Given $(\Lambda, g_z, h_z, g_r, h_u, a, \tilde{a}, b, \tilde{b})$ and uniformly chosen $R_1, \ldots, R_q$, it is hard to find $(z^\star, r^\star, s^\star, t^\star, u^\star, v^\star, w^\star)$ that fulfill relations in (3) under the restriction that $z^\star \neq 1$ and $z^\star \neq z \in R_j$ for every R_j.

Theorem 2. *For any generic algorithm $\mathcal{A}$, the probability that $\mathcal{A}$ breaks SFP with ℓ group operations and pairings is bound by $\mathcal{O}(q^2 + \ell^2)/p$.*

In Appendix A.3, a proof of Theorem 2 is given for the case of $\Lambda = \Lambda_{\mathsf{sym}}$. The argument can be translated to the asymmetric settings. Note that SFP is not covered by the Uber-assumption family of [16]. Instead of extending their framework in a non-trivial way, we show a direct analysis in Appendix A.3. The following holds with respect to SFP and SDP.

Theorem 3. *SFP $\Rightarrow$ SDP.*

A formal proof is in Appendix A.2. An intuition is that, given an answer (z, r, u) to SDP, setting $(s, t, v, w) = (a, \tilde{a}, b, \tilde{b})$ results in a correct answer, (z, r, u, s, t, v, w), to SFP.

2.4 Groth-Sahai Proof System for Pairing Product Equations

The Groth-Sahai (GS) proof system [39] gives efficient non-interactive witness-indistinguishable (NIWI) proofs and non-interactive zero-knowledge (NIZK) proofs for languages that can be described as a set of satisfiable equations, each of which falls in one of the following categories: pairing product equations, multi-exponentiation equations, and general arithmetic gates. The GS proof system could be instantiated under different assumption, in particular the Λ_{sxdh} setting (which is asymmetric pairings for which the the DDH holds both in $\mathbb{G}_1$ and $\mathbb{G}_2$) or under the DLIN assumption in the Λ_{sym} setting.

There are two type of CRS which are computationally indistinguishable: one called "real" which yields perfect soundness and allows extraction if in possession of the trapdoor key, and another "simulated" which yields perfect witness-indistinguishable (WI) proofs which could also be made zero-knowledge (ZK) for some of the equations. When proving a statement, described as a set of equations, the GS proof system first commits to the witness components and then for each equation produces proof elements that the corresponding committed values satisfy the equation.

From the type of equations the GS proof system supports, this paper concerns pairing product equations over variables $x_1, \ldots, x_m \in \mathbb{G}_1$ and $\tilde{x}_1, \ldots, \tilde{x}_n \in \mathbb{G}_2$:

$$\prod_{i=1}^{n} e(g_i, \tilde{x}_i) \prod_{i=1}^{m} e(x_i, \tilde{g}_i) \prod_{i=1}^{m}\prod_{j=1}^{n} e(x_i, \tilde{x}_j)^{c_{i,j}} = T$$

where $\{g_i\}_{i=1}^{n} \in \mathbb{G}_1$, $\{\tilde{g}_i\}_{i=1}^{m} \in \mathbb{G}_2$, $\{c_{i,j}\}_{i=1,j=1}^{m,\ n} \in \mathbb{Z}_p$, and $T \in \mathbb{G}_T$ are constants. When the equations involve variables only in one of the groups, we could use simpler, one-sided, equations which also yield more efficient proofs:

$$\prod_{i=1}^{n} e(g_i, \tilde{x}_i) = T.$$

We use this particular type in Section 9.1.

Also, we note that the GS proof system is only witness indistinguishable (WI) when $T \in \mathbb{G}_T$ is an element in the target group without some particular structure. The WI proof size for a set of q one-sided equations over n variables being satisfiable is $2(n + q)$ when working in the Λ_{sxdh} setting and $3(n+q)$ for Λ_{sym}. If for each equation, the corresponding T is equal to $1_{\mathbb{G}_T}$ or its representation as a pairing product equation is known, the proof system could be transformed into zero-knowledge (ZK) proof system, but this also increases the proof size.

Please refer [39] for further details.

3 Pairing Randomization Techniques

We introduce techniques that randomize elements in a pairing or a pairing product without changing their value in $\mathbb{G}_T$. These useful techniques are used throughout the paper.

- **Inner Randomization** $(x', y') \leftarrow \mathsf{Rand}(x, y)$**:** A pairing $A = e(x, y) \neq 1$ is randomized as follows. Choose $\gamma \leftarrow \mathbb{Z}_p^*$ and let $(x', y') = (x^\gamma, y^{1/\gamma})$. It then holds that (x', y') distributes uniformly over $\mathbb{G}_1 \times \mathbb{G}_2$ under the condition of $A = e(x', y')$. If $A = 1$, then first flip a coin and pick $e(1, 1)$ with probability $1/(2p - 1)$. If it is not selected, flip a coin and pick either $e(1, x)$ or $e(x, 1)$ with probability $1/2$. Then select x uniformly from the corresponding group except for 1.

- **Sequential Randomization** $\{x'_i, y'_i\}_{i=1}^k \leftarrow \mathsf{RandSeq}(\{x_i, y_i\}_{i=1}^k)$**:** A pairing product $A = e(x_1, y_1)\, e(x_2, y_2) \ldots e(x_k, y_k)$ is randomized into $A = e(x'_1, y'_1)\, e(x'_2, y'_2) \ldots e(x'_k, y'_k)$ as follows: Let $(\gamma_1, \ldots, \gamma_{k-1}) \leftarrow \mathbb{Z}_p^{k-1}$. We begin with randomizing the first pairing by using the second pairing as follows. First verify that $y_1 \neq 1$ and $x_2 \neq 1$. If $y_1 = 1$, replace the first pairing $e(x_1, 1)$ with $e(1, y_1)$ with a new random $y_1 (\neq 1)$. The case of $x_2 = 1$ is handled in the same manner. Then multiply $1 = e(x_2^{-\gamma_1}, y_1)\, e(x_2, y_1^{\gamma_1})$ to both sides of the formula. We thus obtain

$$A = e(x_1 x_2^{-\gamma_1}, y_1)\, e(x_2, y_1^{\gamma_1} y_2)\, e(x_3, y_3) \ldots e(x_k, y_k). \tag{4}$$

  Next we randomize the second pairing by using the third one. As before, if $y_1^{\gamma_1} y_2 = 1$ or $x_3 = 1$, replace them to random values. Then multiply $1 = e(x_3^{-\gamma_2}, y_1^{\gamma_1} y_2)\, e(x_3, (y_1^{\gamma_1} y_2)^{\gamma_2})$. We thus have

$$A = e(x_1 x_2^{-\gamma_1}, y_1)\, e(x_2 x_3^{-\gamma_2}, y_1^{\gamma_1} y_2)\, e(x_3, (y_1^{\gamma_1} y_2)^{\gamma_2} y_3) \ldots e(x_k, y_k). \tag{5}$$

  This continues up to the $(k-1)$-st pairing. When done, the value of the i-th pairing distributes uniformly in $\mathbb{G}_T$ due to the uniform choice of γ_i. The k-th pairing follows the distribution determined by A and preceding $k - 1$ pairings. To complete the randomization, every pairing is processed by the inner randomization.

  The sequential randomization can be used to extend a product of k pairings a product of arbitrary $\geq k$ pairings by appending $e(1, 1)$ before randomization. By $\{x'_i, y'_i\}_{i=1}^{k'} \leftarrow \mathsf{Extend}(\{x_i, y_i\}_{i=1}^k)$ for $k' (> k)$ we denote the sequential randomization with extension. Parameters k and k' should be clear from the input and the output.

- **One-side Randomization** $\{x'_i\}_{i=1}^k \leftarrow \mathsf{RandOneSide}(\{g_i, x_i\}_{i=1}^k)$**:** Let g_i be an element in $\mathbb{G}_1^*$ of symmetric setting Λ_{sym}. A pairing product $A = e(g_1, x_1)\, e(g_2, x_2) \ldots e(g_k, x_k)$ is randomized into $A = e(g_1, x'_1)\, e(g_2, x'_2) \ldots e(g_k, x'_k)$ as follows. Let $(\gamma_1, \ldots, \gamma_{k-1}) \leftarrow \mathbb{Z}_p^{k-1}$. First multiply $1 = e(g_1, g_2^{\gamma_1})\, e(g_2, g_1^{-\gamma_1})$ to both sides of the formula. We thus obtain

$$A = e(g_1, x_1 g_2^{\gamma_1})\, e(g_2, x_2 g_1^{-\gamma_1})\, e(g_3, x_3) \ldots e(g_k, x_k). \tag{6}$$

  Next multiply $1 = e(g_2, g_3^{\gamma_2})\, e(g_3, g_2^{-\gamma_2})$. We thus have

$$A = e(g_1, x_1 g_2^{\gamma_1})\, e(g_2, x_2 g_1^{-\gamma_1} g_3^{\gamma_2})\, e(g_3, x_3 g_2^{-\gamma_2}) \ldots e(g_k, x_k). \tag{7}$$

  This continues until γ_{k-1} and we eventually have $A = e(g_1, x'_1) \ldots e(g_k, x'_k)$. Observe that every x'_i for $i = 1, \ldots, k-1$ distributes uniformly in $\mathbb{G}$ due to the uniform multiplicative factor $g_{i+1}^{\gamma_i}$. In the k-th pairing, x'_k follows the distribution determined by A and the preceding $k - 1$ pairings. Thus $(x'_1, \ldots, x'_k)$ is uniform over $\mathbb{G}^k$ under constraint of being evaluated to A.

Note that the algorithms yield uniform elements and thus may include pairings that evaluate to $1_{\mathbb{G}_T}$. If it is not preferable, it can be avoided by repeating that particular step once again excluding the bad randomness.

4 The Main Scheme: Constant-Size Signatures

4.1 Overview

Combining a trapdoor commitment scheme and a strong assumption is a well-known approach for designing signature schemes. To bring this idea into a real construction, we need a trapdoor commitment scheme and a useful (and acceptable) assumption which are compatible with each other, something that is not easily obtained under strong design constraints. In our case, we can build efficient multi-message trapdoor commitment schemes from SDP as shown in Appendix B. Furthermore, SDP is implied by SFP as shown in Theorem 3, so that should allow a smooth combination.

A remaining technical issue is how to deal with "exceptions" such as $z^\star \neq 1$ in SFP. The signature scheme should not inherit it since when proving a knowledge of a signature, the condition $z \neq 1$ is not trivial to prove and affects the efficiency. We address this issue by involving another set of elements $(a_0, \tilde{a}_0)$ and $(b_0, \tilde{b}_0)$ in the verification predicate. In the proof of unforgeability, these elements hold a secret random offset $\tilde{g}^\zeta$ that will be multiplied to z in a forged signature so that the answer to SFP, $z^\star = z\tilde{g}^\zeta$, happens to be 1 only by chance. (The real proof is slightly more involved.)

The randomization techniques from Section 3 also help the construction and the security proof in such a way so that the signature elements are uniform conditioned on satisfying the verification equations.

4.2 Construction

Let $\vec{m} = (m_1, \ldots, m_k) \in \mathbb{G}_2^k$ be a message to be signed. Parameter k determines the length of a message and shorter messages are implicitly padded with $1_{\mathbb{G}_2}$-s. Let $\Lambda \in \{\Lambda_{\mathsf{sym}}, \Lambda_{\mathsf{xdh}}, \Lambda_{\mathsf{sxdh}}\}$. We remind that $\Lambda := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, \tilde{g})$ is an implicit input to the algorithms described below.

- Key Generation. $\mathsf{SIG.Key}(1^\lambda)$: Choose random generators $g_r, h_u \leftarrow \mathbb{G}_1^*$. For $i = 1, \ldots, k$, choose $\gamma_i, \delta_i \leftarrow \mathbb{Z}_p^*$ and compute $g_i = g_r^{\gamma_i}$ and $h_i = h_u^{\delta_i}$. Choose $\gamma_z, \delta_z \leftarrow \mathbb{Z}_p^*$ and compute $g_z = g_r^{\gamma_z}$ and $h_z = h_u^{\delta_z}$. Also choose $\alpha, \beta \leftarrow \mathbb{Z}_p^*$ and compute $\{a_i, \tilde{a}_i\}_{i=0}^1 \leftarrow \mathsf{Extend}(g_r, \tilde{g}^\alpha)$ and $\{b_i, \tilde{b}_i\}_{i=0}^1 \leftarrow \mathsf{Extend}(h_u, \tilde{g}^\beta)$. Set $vk = (g_z, h_z, g_r, h_u, \{g_i, h_i\}_{i=1}^k, \{a_i, \tilde{a}_i, b_i, \tilde{b}_i\}_{i=0}^1)$ and $sk = (vk, \alpha, \beta, \gamma_z, \delta_z, \{\gamma_i, \delta_i\}_{i=1}^k)$. Output (vk, sk).

- Signature Issuing. $\mathsf{SIG.Sign}(sk, \vec{m})$: Choose $\zeta, \rho, \tau, \varphi, \omega$ randomly from $\mathbb{Z}_p$ and set:

$$z = \tilde{g}^\zeta, \quad r = \tilde{g}^{\alpha - \rho\tau - \gamma_z\zeta} \prod_{i=1}^k m_i^{-\gamma_i}, \quad s = g_r^\rho, \quad t = \tilde{g}^\tau,$$
$$u = \tilde{g}^{\beta - \varphi\omega - \delta_z\zeta} \prod_{i=1}^k m_i^{-\delta_i}, \quad v = h_u^\varphi, \quad w = \tilde{g}^\omega.$$

  Output $\sigma = (z, r, s, t, u, v, w)$ as a signature.

- Verification. $\mathsf{SIG.Vrf}(vk, \vec{m}, \sigma)$: Parse σ into (z, r, s, t, u, v, w). Output 1 if

$$A = e(g_z, z)\; e(g_r, r)\; e(s, t)\; \prod_{i=1}^{k} e(g_i, m_i), \text{ and} \tag{8}$$

$$B = e(h_z, z)\; e(h_u, u)\; e(v, w)\; \prod_{i=1}^{k} e(h_i, m_i) \tag{9}$$

hold for $A = e(a_0, \tilde{a}_0)\; e(a_1, \tilde{a}_1)$ and $B = e(b_0, \tilde{b}_0)\; e(b_1, \tilde{b}_1)$. Output 0, otherwise.

4.3 Security

Theorem 4. *SIG in Section 4.2 is correct. It is EUF-CMA if SFP holds for Λ.*

Proof. CORRECTNESS. Observe that

$$e(g_z, z)\; e(g_r, r)\; e(s, t)\; \prod_{i=1}^{k} e(g_i, m_i) \;=$$

$$= e\left(g_r^{\gamma_z}, \tilde{g}^{\zeta}\right)\; e\left(g_r, \tilde{g}^{\alpha - \rho\tau - \gamma_z\zeta} \prod_{i=1}^{k} m_i^{-\gamma_i}\right)\; e\left(g_r^{\rho}, \tilde{g}^{\tau}\right) \prod_{i=1}^{k} e\left(g_r^{\gamma_i}, m_i\right) = \; e\left(g_r, \tilde{g}^{\alpha}\right) \;=\; A$$

holds. Thus (8) is fulfilled. Relation (9) is verified in the same manner.

UNFORGEABILITY. Let $\mathcal{A}$ be an adversary that has a non-negligible advantage of forging a signature for the above scheme on a message $\vec{m}^{\dagger}$, $\vec{m}^{\dagger} \notin \{\vec{m}_j\}_{j=1}^{q}$, after adaptively querying the signing oracle on messages $\vec{m}_j$, for $j = 1, \ldots, q$, and receiving signatures σ_j. We construct a reduction algorithm which takes an input Λ, g_z, h_z, g_r, h_u, $(a, \tilde{a}), (b, \tilde{b})$, and uniformly chosen tuples R_j for $j = 1, \ldots, q$ as defined in Assumption 3, and simulates the view of $\mathcal{A}$ in the attack environment as follows:

- (Simulating $\mathsf{SIG.Key}$) : Use (g_z, h_z, g_r, h_u) as given in the input. For $i = 1, \ldots, k$ set $g_i = g_z^{\chi_i} g_r^{\gamma_i}$ and $h_i = h_z^{\chi_i} h_u^{\delta_i}$, where $\chi_i, \gamma_i,\ \delta_i \leftarrow \mathbb{Z}_p$. As the probability that any g_i or h_i, $i = 1, \ldots, k$, is equal to $1_{\mathbb{G}_1}$ is negligible, the reduction algorithm simply aborts in such cases. Otherwise, all group elements are from $\mathbb{G}_1^*$ and chosen uniformly at random, like in the key generation algorithm. Then select $\zeta,\ \rho,\ \varphi \leftarrow \mathbb{Z}_p$, and compute $((a_0, \tilde{a}_0), (a_1, \tilde{a}_1)) \leftarrow \mathsf{RandSeq}((g_z^{\zeta} g_r^{\rho}, \tilde{g}), (a, \tilde{a}))$ and $((b_0, \tilde{b}_0), (b_1, \tilde{b}_1)) \leftarrow \mathsf{RandSeq}((h_z^{\zeta} h_u^{\varphi}, \tilde{g}), (b, \tilde{b}))$. For convenience, denote $g_z^{\zeta} g_r^{\rho}$ with a' and $h_z^{\zeta} h_u^{\varphi}$ with b'. The verification key is $vk = (g_z, h_z, g_r, h_u, \{g_i, h_i\}_{i=1}^{k}, \{a_i, \tilde{a}_i, b_i, \tilde{b}_i\}_{i=0}^{1})$.

- (Simulating $\mathsf{SIG.Sign}$) : Given message $\vec{m}$, take a fresh tuple $R_j = (z_j, r_j, s_j, t_j, u_j, v_j, w_j)$ from the input instance. Then compute

$$z = z_j\; \tilde{g}^{\zeta} \prod_{i=1}^{k} m_i^{-\chi_i}, \quad r = r_j\; \tilde{g}^{\rho} \prod_{i=1}^{k} m_i^{-\gamma_i}, \quad s = s_j, \quad t = t_j,$$

$$u = u_j\; \tilde{g}^{\varphi} \prod_{i=1}^{k} m_i^{-\delta_i}, \quad v = v_j, \quad w = w_j.$$

The signature is $\sigma = (z, r, s, t, u, v, w)$. It is easy to verify that the signature satisfies the verification equations.

When $\mathcal{A}$ outputs $(\vec{m}^\dagger, (z^\dagger, r^\dagger, s^\dagger, t^\dagger, u^\dagger, v^\dagger, w^\dagger))$, compute

$$z^\star = z^\dagger \, \tilde{g}^{-\zeta} \prod_{i=1}^{k} \left(m_i^\dagger\right)^{\chi_i}, \quad r^\star = r^\dagger \, \tilde{g}^{-\rho} \prod_{i=1}^{k} \left(m_i^\dagger\right)^{\gamma_i}, \quad u^\star = u^\dagger \, \tilde{g}^{-\varphi} \prod_{i=1}^{k} \left(m_i^\dagger\right)^{\delta_i},$$

and set $s^\star = s^\dagger$, $t^\star = t^\dagger$, $v^\star = v^\dagger$, and $w^\star = w^\dagger$. If any of the parameters $\chi_1, \ldots, \chi_k$ is 0 the reduction algorithm aborts; otherwise, outputs $(z^\star, r^\star, s^\star, t^\star, u^\star, v^\star, w^\star)$. Like in the previous abort case, the chance for that is negligible because the parameters are chosen uniformly at random, and, therefore, we could ignore those cases in our analysis without affecting the overall outcome. This completes the description of the reduction algorithm.

The above signatures follow correct distribution. So, $\mathcal{A}$ outputs a successful forgery with a non-negligible probability. Then, for the output of the reduction algorithm, it holds that

$$\begin{aligned} e(g_z, z^\star)\, e(g_r, r^\star)\, e(s^\star, t^\star) &= e\left(g_z, z^\dagger \, \tilde{g}^{-\zeta} \prod_{i=1}^{k} \left(m_i^\dagger\right)^{\chi_i}\right) \, e\left(g_r, r^\dagger \, \tilde{g}^{-\rho} \prod_{i=1}^{k} \left(m_i^\dagger\right)^{\gamma_i}\right) \, e\left(s^\dagger, t^\dagger\right) \\ &= e\left(g_z^{-\zeta} g_r^{-\rho}, \tilde{g}\right) \, e\left(g_z, z^\dagger\right) \, e\left(g_r, r^\dagger\right) \, e\left(s^\dagger, t^\dagger\right) \, \prod_{i=1}^{k} e\left(g_i, m_i^\dagger\right) \\ &= e\left(g_z^{\zeta} g_r^{\rho}, \tilde{g}\right)^{-1} \, \prod_{i=0}^{1} e(a_i, \tilde{a}_i) = e(a, \tilde{a}). \end{aligned}$$

One can also verify that $e(g_z, z^\star)\, e(h_u, u^\star)\, e(v^\star, w^\star) = e(b, \tilde{b})$ holds in the same way.

What remains is to show that $z^\star$ is not in $\{1, z_1, \ldots, z_q\}$. For that, first notice that the parameters ζ and $\{\chi_i\}_{i=1}^{k}$ are independent from the view of adversary $\mathcal{A}$, as proved in Lemma 1. Namely, for any view of the adversary and for any choice of ζ and χ_i, for $i = 1, \ldots, k$, there exist unique and consistent parameters ρ, φ, γ_i, δ_i, $i = 1, \ldots, k$ and z_j, r_j, u_j, $j = 1, \ldots, q$.

First we show that the probability $z^\star \in \{z_1, \ldots, z_q\}$ is negligible. For every z_j and signature $\sigma = (z, r, s, t, u, v, w)$ on a message $\vec{m}$ simulated by using z_j, it holds that

$$\frac{z^\star}{z_j} \quad = \quad \frac{z^\dagger \, \tilde{g}^{-\zeta} \prod_{i=1}^{k} \left(m_i^\dagger\right)^{\chi_i}}{z \, \tilde{g}^{-\zeta} \, \prod_{i=1}^{k} m_i^{\chi_i}} \quad = \quad \frac{z^\dagger}{z} \prod_{i=1}^{k} \left(\frac{m_i^\dagger}{m_i}\right)^{\chi_i}.$$

Since $\vec{m}^\dagger \neq \vec{m}$, there exists i such that $m_i^\dagger \neq m_i$. Since $\chi_i \in \mathbb{Z}_p^*$ is information theoretically hidden from the view of the adversary, the probability that $z^\star = z_j$ is negligible due to the term $(m_i^\dagger / m_i)^{\chi_i}$ in the above equation. To show that $z^\star = \left(z^\dagger\right) \tilde{g}^{-\zeta} \prod_{i=1}^{k} \left(m_i^\dagger\right)^{\chi_i}$ is equal to $1_{\mathbb{G}_2}$ only with a negligible probability, notice that ζ is also independent from the view of the adversary and the claim holds due to the uniform choice of ζ. Therefore, when the reduction algorithm does not abort, the probability that $z^\star \notin \{1, z_1, \ldots, z_q\}$ is overwhelming. ∎

Lemma 1. *The parameters ζ, χ_1, χ_2, …, χ_k chosen by the reduction algorithm in Theorem 4 are independent from $\mathcal{A}$'s view. That is, independent from the verification key, the signed messages, and the signatures.*

Proof. Let $vk = (g_z, h_z, g_r, h_u, \{g_i, h_i\}_{i=1}^{k}, \{a_i, \tilde{a}_i, b_i, \tilde{b}_i\}_{i=0}^{1})$ be the verification key the adversary sees, $\vec{m}_1$, …, $\vec{m}_q$ be the messages with which $\mathcal{A}$ queries the signing oracle, and σ_1, …, σ_q be the corresponding signatures. Furthermore, let assume that $(a, \tilde{a})$ and $(b, \tilde{b})$ given to the reduction algorithm are also fixed, though $\mathcal{A}$ does not see them. That yields unique a' and b' such that

$$A = e(a_0, \tilde{a}_0)e(a_1, \tilde{a}_1) = e(a', \tilde{g})e(a, \tilde{a}) \quad \text{and} \quad B = e(b_0, \tilde{b}_0)e(b_1, \tilde{b}_1) = e(b', \tilde{g})e(b, \tilde{b}).$$

For any choice $\hat{\zeta}, \hat{\chi}_i \in \mathbb{Z}_p$ of the parameters ζ, χ_i , for $i = 1, \ldots, k$, there exist a *unique* coin toss $\hat{\rho}$, $\hat{\varphi}$, $\hat{\gamma}_i$, $\hat{\delta}_i$ such that $a' = g_z^{\hat{\zeta}} g_r^{\hat{\rho}}$, $b' = h_z^{\hat{\zeta}} h_u^{\hat{\varphi}}$, $g_i = g_z^{\hat{\chi}_i} g_r^{\hat{\gamma}_i}$, and $h_i = h_z^{\hat{\chi}_i} h_u^{\hat{\delta}_i}$. This shows that the verification key and the parameters are independent. Next we show that the chosen parameters remain independent from $\mathcal{A}$'s view even after signing q adaptively chosen messages due to the uniform choice of the tuples R_j, $j = 1, \ldots, q$, as defined in Assumption 3.

Let the j-th message be $\vec{m}$ and the corresponding signature be $\sigma = (z, r, s, t, u, v, w)$. From the specification of the reduction algorithm we know that $(s, t) = (s_j, t_j)$ and $(v, w) = (v_j, w_j)$, where $R_j = (z_j, r_j, s_j, t_j, u_j, v_j, w_j)$ is the j-th tuple given as input. And for the fixed view, $\zeta, \{\chi_i\}_{i=1}^k$ determine *uniquely* the values of $z_j = z\ \tilde{g}^{-\zeta} \prod_{i=1}^k m_i^{\chi_i}$, $r_j = r\ \tilde{g}^{-\rho} \prod_{i=1}^k m_i^{\gamma_i}$, and $u_j = u\ \tilde{g}^{-\varphi} \prod_{i=1}^k m_i^{\delta_i}$. Regardless of the particular choice of parameters $\hat{\zeta}, \{\hat{\chi}_i\}_{i=1}^k$, since σ satisfies the signature verification equations:

$$A = e(g_z, z)\ e(g_r, r)\ e(s, t) \prod_{i=1}^k e(g_i, m_i) \quad \text{and} \quad B = e(h_z, z)\ e(h_u, u)\ e(v, w) \prod_{i=1}^k e(h_i, m_i),$$

it is true that the corresponding tuple $\hat{R}_j = (\hat{z}_j, \hat{r}_j, s_j, t_j, \hat{u}_j, v_j, w_j)$ satisfies:

$$e(a, \tilde{a}) = e(g_z, \hat{z}_j)\ e(g_r, \hat{r}_j)\ e(s_j, t_j) \quad \text{and} \quad e(b, \tilde{b}) = e(h_z, \hat{z}_j)\ e(h_u, \hat{u}_j)\ e(v_j, w_j). \tag{10}$$

What remains to show is that the uniform choice of $\hat{\zeta}, \{\hat{\chi}_i\}_{i=1}^k$ together with $\mathcal{A}$'s view yields uniform distribution for the tuples $\hat{R}_j$, for $j = 1, \ldots, q$, as specified by the assumption description. If that is indeed the case, each set of tuples which could have been given as input to the reduction algorithm is chosen with the same probability. And because for any choice of $\hat{\zeta}, \hat{\chi}_1, \ldots, \chi_k$, there exist *unique* set $\{\hat{R}_j\}_{j=1}^q$, those imply that each parameter selection looks equally likely for $\mathcal{A}$.

To see the uniformity of $\hat{R}_j$, note again that (s_j, t_j) and (v_j, w_j) are determined uniquely from the view regardless of the parameters choice. Then, let's define the homomorphism ϕ:

$$\phi_{\tilde{g}, \vec{m}}(\hat{\zeta}, \hat{\chi}_1, \ldots, \hat{\chi}_k) = \tilde{g}^{-\hat{\zeta}} \prod_{i=1}^k m_i^{\hat{\chi}_i}.$$

It is easy to verify that for uniformly chosen parameters, the homomorphism's range is uniformly distributed over $\mathbb{G}_2$. This in turn implies that for a fixed z and uniformly chosen parameters, $\hat{z}_j = z\ \phi(\hat{\zeta}, \chi_1, \ldots, \chi_k)$ is uniformly distributed over $\mathbb{G}_2$. And because $\hat{R}_j$ satisfies (10), the values of $\hat{r}_j$ and $\hat{u}_j$ are determined uniquely by the other tuple values, which for a fixed view means determined by $\hat{z}_j$. To sum it up, for a fixed view, the uniform random choice of the parameters gives uniformly distributed $\hat{z}_j$ which implies the uniformity of $\hat{R}_j$.

∎

4.4 Notable Properties

Partial Perfect Randomizability. Given a signature (z, r, s, t, u, v, w) one can randomize every element except for z by applying the sequential randomization technique with small tweak as follows. Define the function $\mathsf{SigRand}$, $(r', s', t', u', v', w') \leftarrow \mathsf{SigRand}(r, s, t, u, v, w)$, as:

- Randomize (r, s, t) into (r', s', t') as follows.
  - First, if $t = 1$, set $s = 1$ and choose $t \leftarrow \mathbb{G}_2^*$.
  - Then, choose $\varrho \leftarrow \mathbb{Z}_p$ and compute
$$r' = r\, t^{\varrho}, \quad (s', t') \leftarrow \mathsf{Rand}(s g_r^{-\varrho}, t) \tag{11}$$
- Randomize (u, s, t) into (u', s', t') analogously.

Lemma 2. *The above (r', s', t', u', v', w') distributes uniformly over $(\mathbb{G}_2 \times \mathbb{G}_1 \times \mathbb{G}_2)^2$ under constraint that $e(g_r, r)\, e(s, t) = e(g_r, r')\, e(s', t')$ and $e(h_u, u)\, e(v, w) = e(h_u, u')\, e(v', w')$.*

Proof. Uniformity of $r' \in \mathbb{G}_2$ follows from $t \neq 1$ and the uniformity of ϱ in (11). Under the described constraints, for any choice of r', their is a *unique* value $e(s', t') = e(g_r, r)\, e(s, t)\, e(g_r, r')^{-1}$. Then, uniformity of s' and t' holds from the property of Rand. The same is true for (u', v', w'). ∎

The claim implies that (s', t', v', w') is information theoretically independent of the signature element z, the message, and the verification key. (In general, the same is true for publishing any two elements from (r', s', t') and (u', v', w') respectively.) This property is useful in reducing the task of combined proofs. See Section 9.1 for typical use of this property.

Signature Binding Property. Roughly, it claims that no one but the signer can obtain two signatures which have the same s and v. In the following formal statement, the adversary is allowed to submit both $\vec{m}$ and $\vec{m}'$ to the signing oracle. Hence the property is not implied by EUF-CMA in general.

Lemma 3. *Under adaptive chosen message attacks, no adversary can output $(\vec{m}, \sigma)$ and $(\vec{m}', \sigma')$ such that $1 = \mathsf{SIG.Vrf}(vk, \vec{m}, \sigma) = \mathsf{SIG.Vrf}(vk, \vec{m}', \sigma')$, $\vec{m} \neq \vec{m}'$, and (s, v) are shared in σ and σ'.*

Lemma 3 implies that publishing (s, v) together with the verification key works as a commitment of the signature and the message. (Recall that s and v are uniformly chosen in the signature generation algorithm.) This property is used in Section 5, and would find more applications.

Proof. Suppose that there is a successful adversary, $\mathcal{A}$ that outputs the signatures as in the lemma. We then construct an adversary $\mathcal{B}$ that breaks EUF-CMA of SIG.

Given vk and oracle access to $\mathcal{O}_{\mathsf{sign}}$, $\mathcal{B}$ invokes $\mathcal{A}$ with vk. Every signing query from $\mathcal{A}$ is directly passed to $\mathcal{O}_{\mathsf{sign}}$ and the signatures are returned directly to $\mathcal{A}$. Hence $\mathcal{B}$'s simulation is perfect. Eventually, $\mathcal{A}$ terminates and outputs $\sigma = (z, r, s, t, u, v, w)$, $\vec{m} = (m_1, \ldots, m_k)$, $\sigma' = (z', r', s, t', u', v, w')$, and $\vec{m}' = (m'_1, \ldots, m'_k)$.

$\mathcal{B}$ then chooses $\varrho \leftarrow \mathbb{Z}_p$ and computes
$$z^\star = z \left(\tfrac{z'}{z}\right)^{\varrho}, \quad r^\star = r \left(\tfrac{r'}{r}\right)^{\varrho}, \quad t^\star = t \left(\tfrac{t'}{t}\right)^{\varrho}, \quad u^\star = u \left(\tfrac{u'}{u}\right)^{\varrho}, \quad w^\star = w \left(\tfrac{w'}{w}\right)^{\varrho}, \quad m_i^\star = m_i \left(\tfrac{m'_i}{m_i}\right)^{\varrho}.$$
Then outputs $\sigma^\star = (z^\star, r^\star, s, t^\star, u^\star, v, w^\star)$ and $\vec{m}^\star = (m_1^\star, \ldots, m_k^\star)$. This completes the specification of $\mathcal{B}$.

We verify the correctness of $\mathcal{B}$ as follows. Since these signatures are valid, they satisfy
$$A = e(g_z, z')\, e(g_r, r')\, e(s, t') \prod_{i=1}^{k} e(g_i, m'_i) = e(g_z, z)\, e(g_r, r)\, e(s, t) \prod_{i=1}^{k} e(g_i, m_i), \text{ and} \tag{12}$$
$$B = e(h_z, z')\, e(h_u, u')\, e(v, w') \prod_{i=1}^{k} e(h_i, m'_i) = e(h_z, z)\, e(h_u, u)\, e(v, w) \prod_{i=1}^{k} e(h_i, m_i). \tag{13}$$

Note that we can divide the verification equations of the signatures which gives us:

$$1_{\mathbb{G}_T} = e\left(g_z, \frac{z'}{z}\right)\, e\left(g_r, \frac{r'}{r}\right)\, e\left(s, \frac{t'}{t}\right) \prod_{i=1}^{k} e\left(g_i, \frac{m'_i}{m_i}\right), \text{ and}$$

$$1_{\mathbb{G}_T} = e\left(h_z, \frac{z'}{z}\right)\, e\left(h_u, \frac{u'}{u}\right)\, e\left(v, \frac{w'}{w}\right) \prod_{i=1}^{k} e\left(h_i, \frac{m'_i}{m_i}\right).$$

Exponentiating these equations with ϱ and multiplying them with one of the signatures yields $\sigma^\star = (z^\star, r^\star, s, t^\star, u^\star, v, w^\star)$ and $\vec{m}^\star = (m_1^\star, \ldots, m_k^\star)$ which clearly satisfy the verification equations.

Since $\vec{m}' \neq \vec{m}$ there exists j such that $m'_j \neq m_j$. And due to the random choice of ϱ, $m_j^\star = m_j \left(\frac{m'_j}{m_i}\right)^{\varrho}$ distributes uniformly over $\mathbb{G}_2$. Accordingly, $\vec{m}^\star$ is different from any message vector observed by $\mathcal{O}_{\mathsf{sign}}$ with overwhelming probability. Thus, $(\sigma^\star, \vec{m}^\star)$ is a valid forgery to SIG. ▮

4.5 Variations

- We can replace $a_i, \tilde{a}_i, b_i, \tilde{b}_i$ with $A = e(g_r, \tilde{g}^\alpha)$ and $B = e(h_u, \tilde{g}^\beta)$ in a verification-key, and use the A and B directly in the verification equations (8) and (9). The reason we include a representation of A (and B) in $\mathbb{G}_1$ and $\mathbb{G}_2$ is to address the needs to put the verification key into the base groups. The GS-proof system provides zero-knowledge property for statements that do not include elements from $\mathbb{G}_T$ except for $1_{\mathbb{G}_T}$. When WI is of only concern, one can include A and B in vk and use them directly in the verification. We use this modification in Section 9.1. The same is possible for other schemes in this paper.
- Let $\langle n \rangle$ denote a deterministic encoding of non-negative integer n $(< p)$ to an element of $\mathbb{G}_2^*$. By limiting the maximum message length to be $k-1$ and putting $\langle |\vec{m}| \rangle$ at the beginning of the input message $\vec{m}$, shorter messages can be treated. Since the encoding is deterministic and black-box that is independent of the representation of the elements in $\vec{m}$, it does not impact the compatibility.
- As we observed in the very last stage of the security proof, $(a_0, \tilde{a}_0)$ and $(b_0, \tilde{b}_0)$ in a verification key is needed to handle the case where $z^\dagger = 1$ and $\vec{m}^\dagger = (1, \ldots, 1)$ happen at the same time. When such exception is not possible, for example when $\vec{m}$ is encoded with with its length as $\vec{m}^\dagger = (\langle n \rangle, 1, \ldots, 1)$ and the deterministic encoding $\langle n \rangle$ is never 1, the elements $(a_0, \tilde{a}_0)$ and $(b_0, \tilde{b}_0)$ can be removed from the scheme.
- In the asymmetric settings, one can swap $\mathbb{G}_1$ and $\mathbb{G}_2$ in the description of SIG to get the 'dual' scheme of SIG whose message space is $\mathbb{G}_1^k$.
- Dropping the flexible part $e(s,t)$ and $e(v,w)$ from the construction results in a strongly unforgeable one-time signature scheme based on the SDP assumption. Also, the verification-key size can be reduced a little. See Appendix C for details.

5 Signing Unbounded-Size Messages

5.1 Overview

This section presents a method to sign message $(m_1, \ldots, m_n)$ whose size n is not a-priori bounded by the public-key. While some generic domain extension methods are available, we present a specific and efficient construction based on a *chain of signatures* taking the advantage of the constant-size signature scheme from Section 4. The idea is that, first sign m_1 to obtain σ_1, and next sign $\sigma_1||m_2$ to obtain σ_2, then sign $\sigma_2||m_3$ and so on. (Note that this rough description lacks some important details. In particular, signing only on m_1 at the beginning results in an insecure scheme.)

A technical highlight is that, with our constant-size signature scheme, we only need to involve a part of a signature, elements s and v, in each step of chaining to constitute a secure chain. This is possible due to the signature binding property of SIG as shown in Section 4.4.

5.2 Construction

Let SIG be the constant-size signature scheme from Section 4, whose message space is $\mathbb{G}_2^k$ for $k \geq 3$. We construct an unbounded-message signature scheme, $\mathsf{USIG1}$, as follows. Let $\Lambda = \Lambda_{\mathsf{sym}}$ be implicitly given to the functions described below. Recall that $\langle n \rangle$ is an encoding of n to an element of $\mathbb{G}_2^*$.

- $\mathsf{USIG1.Key}(1^\lambda)$: Generate random $(s_{-1}, v_{-1}) \in \mathbb{G}_1^2$. Invoke $(vk', sk) \leftarrow \mathsf{SIG.Key}(1^\lambda)$. Output $vk = (vk', s_{-1}, v_{-1})$ and sk.

- $\mathsf{USIG1.Sign}(sk, \vec{m})$: Parse $\vec{m}$ into $(m_1, \ldots, m_n)$. Let $\ell = \lceil \frac{n+1}{k-2} \rceil$. Let $m_0 = \langle n \rangle$ and $m_i = 1$ for $i = n+1, \ldots, \ell(k-2)$. For $i = 0, \ldots, \ell - 1$, compute $\sigma_i = (z_i, r_i, s_i, t_i, u_i, v_i, w_i) \leftarrow \mathsf{SIG.Sign}(sk, \vec{m}_i)$ where $\vec{m}_i = (s_{i-1}, v_{i-1}, m_{i(k-2)}, \ldots, m_{(i+1)(k-2)-1})$. Output $\sigma = (\sigma_0, \ldots, \sigma_{\ell-1})$.

- $\mathsf{USIG1.Vrf}(vk, \vec{m}, \sigma)$: Parse σ into $(\sigma_0, \ldots, \sigma_{\ell-1})$ and $\vec{m}$ into $(m_1, \ldots, m_n)$. Let $m_0 = \langle n \rangle$ and $m_i = 1$ for $i = n+1, \ldots, \ell(k-2)$. For $i = 0, \ldots, \ell - 1$, compute $b_i = \mathsf{SIG.Vrf}(vk', \vec{m}_i, \sigma_i)$ where $\vec{m}_i$ is formed in the same way as in $\mathsf{SIG.Sign}$. Output 1 if $b_i = 1$ for all $i = 0, \ldots, \ell - 1$. Output 0, otherwise.

The resulting signature is in the size of $7 \cdot \lceil \frac{n+1}{k-2} \rceil$.

Remarks. Filling $1_{\mathbb{G}}$ to the sloppy slots of the message space is for notational consistency. It does not increase either computation or storage. Setting $\Lambda = \Lambda_{\mathsf{sym}}$ is needed as (s_i, v_i) is in $\mathbb{G}_1^2$ while the message space is $\mathbb{G}_2^k$. It can be modified for the case of $\Lambda = \Lambda_{\mathsf{sxdh}}$ using the signature scheme described in Section 7 (but not for the case of $\Lambda = \Lambda_{\mathsf{xdh}}$). If $\vec{m}$ is given as an on-line stream and the length is not known in advance, one can use the trapdoor commitment scheme $\mathsf{TC2}$ from Appendix B so that m_0 is set to a random commitment and later opened to n when n is fixed. The opening information is included as a part of a signature. Since the opening information is a group element and the commitment verification predicate is a pairing product equation, the resulting verification predicate for $\mathsf{USIG1}$ remains as a conjunction of pairing product equations.

Theorem 5. *If SIG is EUF-CMA, so is $\mathsf{USIG1}$.*

Proof. Suppose that there is a successful adversary, say $\mathcal{A}$, that launches chosen message attacks and outputs a valid forgery, $((m_1^\dagger, \ldots, m_n^\dagger), (\sigma_0^\dagger, \ldots, \sigma_{\ell-1}^\dagger))$. Let $\vec{m}_i^\dagger$ be the message vector associated to $\sigma_i^\dagger$. We then have two cases.

Type-I. There is $\vec{m}_i^\dagger$ that has never been signed by the signing oracle.

Type-II. Every $\vec{m}_i^\dagger$ has been signed by the signing oracle (in separate queries).

Type-I forgery trivially breaks the unforgeability of SIG. For Type-II forgery, we show a reduction to the unforgeability of SIG as follows. Given verification key vk' of SIG and access to the signing oracle of SIG, we construct a simulator that uses adversary $\mathcal{A}$ and simulates $\mathsf{USIG1}$ as follows. Let $\mathcal{O}_{\mathsf{sign}}$ be the signing oracle of SIG with respect to vk'.

- (Simulating $\mathsf{USIG1.Key}$): Generate a random message vector $\vec{m}_{-1}$ of size k and send it to $\mathcal{O}_{\mathsf{sign}}$. Receive signature $(z_{-1}, r_{-1}, s_{-1}, t_{-1}, u_{-1}, v_{-1}, w_{-1})$ and output $vk = (vk', s_{-1}, v_{-1})$.
- (Simulating $\mathsf{USIG1.Sign}$): On input $\vec{m}$, follow the legitimate signing algorithm by asking $\mathcal{O}_{\mathsf{sign}}$ to compute $\mathsf{SIG.Sign}$. Then output the resulting signature.

Observe that s_{-1} and v_{-1} generated in the simulated $\mathsf{USIG1.Key}$ are uniform and independent of $\vec{m}_{-1}$. Simulation for $\mathsf{USIG1.Sign}$ is clearly perfect as it follows the legitimate procedure.

Suppose that adversary $\mathcal{A}$ outputs a valid forgery for $\mathsf{USIG1}$. Then there exists a signing query (to the signing oracle of $\mathsf{USIG1}$) in which $\vec{m}^{\dagger}_{\ell-1}$ is observed. Let $((m_1, \ldots, m_{n'}), (\sigma_0, \ldots, \sigma_{\ell'-1}))$ be the message and the signature with respect to the query and let $\vec{m}_i$ denote a message vector associated to σ_i. Let $i^\star$ be the index where $\vec{m}^{\dagger}_{\ell-1} = \vec{m}_{i^\star}$ happens. If $\ell - 1 = 0$, then $i^\star = 0$ is not the case because the message in the valid forgery must be fresh. In the case of $\ell - 1 \neq 0$ and $i^\star = 0$, it happens that $\vec{m}^{\dagger}_{\ell-2} \neq \vec{m}_{-1}$ with overwhelming probability since $\vec{m}_{-1}$ is chosen randomly and information theoretically independent from the view of the adversary. The same is true for the case of $\ell - 1 = 0$ and $i^\star > 0$. In the case of $\ell - 1 \neq 0$ and $i^\star > 0$, since the messages are prefix-free, there exists $j^\star$ such that $\vec{m}^{\dagger}_{\ell-1-j^\star} \neq \vec{m}_{i^\star - j^\star}$ happens for the first time when $j^\star$ is increased from 0 to $\min(\ell - 1, i^\star) + 1$. In any of the cases ($j^\star$ is set to 1 for the case of $i^\star = 0$ or $\ell - 1 = 0$), signature $\sigma^{\dagger}_{\ell-1-j^\star}$ shares s and v with $\sigma_{i^\star - j^\star}$ as they are included in $\vec{m}_{i^\star - j^\star + 1} (= \vec{m}^{\dagger}_{\ell-1-j^\star+1})$. This contradicts to the signature binding property of SIG as claimed in Lemma 3. ∎

6 Simulatable Signatures

6.1 Overview

A *simulatable signature scheme* is a signature scheme in the CRS model that allows to create valid signatures without the signing-key but with a trapdoor associated to the common reference string. The notion is introduced in [3] but in an informal way dedicated for their purposes. We elaborate the notion and present a formal treatment with reasonable construction in this section.

A simulatable signature is a useful tool in combination with a witness indistinguishable (WI) proof system. Unlike zero-knowledge (ZK) proofs, WI proof system does not accompany a simulator. So when a signature is a part of the witness and the signer is corrupt and useless, simulatable signature can provide a correct witness to the entity having the trapdoor. This situation happens in reality, for instance, when we attempt to instantiate Fischlin's round-optimal blind signature scheme [28] (modified to use WI as suggested in [40, 3]).

It is known that a simulatable signature scheme can be unconditionally constructed from any regular signature scheme by modifying the verification predicate in such a way that a signature is accepted if it passes regular verification with respect to the signer's verification key *or* the verification key included in the CRS. This generic construction, however, inherently involves disjunction in the resulting verification predicate.

Our construction shares the idea of two-keys. But we use a trapdoor commitment scheme and a signature scheme combined. We assign a commitment-key to the CRS and use a signing-key for real signature generation. Then a reference signature on a default message is included into a verification key. When simulation is needed, we use the trapdoor for the commitment scheme and equivocate the reference signature to be valid with a given message. Since our main scheme in Section 4 already integrate a trapdoor commitment scheme in its construction, it would seem possible to move the commitment part of the verification key into the CRS. And we mostly follow this way. A formal proof however reveals that we need to have k flexible pairings to sign messages of size k, $k \geq 1$, without needing the trapdoor for the commitment part. This results in relying on k-SFP rather than SFP when dealing with messages of size $k \geq 2$.

6.2 Definitions

Definition 3 (Simulatable Signature Scheme). A simulatable signature scheme SSIG consists of algorithms $\mathsf{SSIG}.\{\mathsf{Crs}, \mathsf{Key}, \mathsf{Chk}, \mathsf{Sign}, \mathsf{Vrf}, \mathsf{Sim}\}$ where $\mathsf{SSIG}.\{\mathsf{Key}, \mathsf{Sign}, \mathsf{Vrf}\}$ constitute a regular signature scheme (except that they take the CRS), and the extra algorithms works as follows.

$\mathsf{SSIG.Crs}(1^\lambda)$: A CRS generation algorithm that, on input security parameter λ, outputs a common reference string Σ and a trapdoor τ.

$\mathsf{SSIG.Chk}(\Sigma, vk)$: A verification key checking algorithm that, on input a verification key, returns 1 or 0.

$\mathsf{SSIG.Sim}(\Sigma, vk, m, \tau)$: A signature simulation algorithm that computes a signature σ for message m by using trapdoor τ.

By $\mathcal{M}_{vk}$, we denote the message space associated to vk. By $\mathcal{K}$, we denote the set of (vk, sk) that can be generated by $\mathsf{SSIG.Key}(\Sigma)$. Also by $\mathcal{S}_{sk,m}$ we denote the set of signatures that can be generated by $\mathsf{SSIG.Sign}(\Sigma, sk, m)$.

Completeness is defined in a standard way; with respect to correctly generated CRS, verification keys, and signatures, the verification function outputs 1 with probability 1.

Signature simulatability is defined in such a way that whenever adversary selects an appropriate message and verification key, then, by using the trapdoor of the CRS, it is possible to generate a signature that could have been generated by the proper signing operation. Formal definition follows.

Definition 4 (Signature-Simulatability). A signature scheme in the CRS model is simulatable if, for every CRS Σ generated by $(\Sigma, \tau) \leftarrow \mathsf{SSIG.Crs}(1^\lambda)$, for any (m, vk), if $1 = \mathsf{SSIG.Chk}(\Sigma, vk) \wedge m \in \mathcal{M}_{vk}$, then there exists sk such that $(vk, sk) \in \mathcal{K}$, and $1 = \mathsf{SSIG.Vrf}(\Sigma, vk, m, \sigma)$ holds for any $\sigma \leftarrow \mathsf{SSIG.Sim}(\Sigma, vk, m, \tau)$.

A relaxation would allow a negligible error in $\mathsf{SSIG.Vrf}$ for a message and a verification key chosen by an adversary. Note that the signature simulatability does not require simulated signatures be indistinguishable from the real ones. It is considered as a role of witness indistinguishable proof system coupled with the signature scheme.

Unforgeability is defined with respect to adaptive chosen message attacks. In the CRS model, however, a CRS is used for generating many keys and therefore, we must be careful that the keys should not be badly affected each other. By reflecting this concern, we allow an adversary to access an oracle that outputs correctly generated verification keys with respect to the same CRS. Furthermore, in our potential applications, the adversary is given a witness indistinguishable proof of holding a correct signature with respect to a given message and verification key. Let $\pi \leftarrow \mathsf{NIWI.Prf}((\Sigma, vk_i, m), \sigma)$ denote the proof system for this purpose. Here (Σ, vk_i, m) is public, and σ is the witness, and π is the proof. We do not give much details to the proof system as only the property needed in this formulation is the witness indistinguishability. The CRS for this proof system is implicitly given to the adversary. In summary the attack model includes the following three oracles.

- (Key Generation Oracle $\mathcal{O}_{\mathsf{vk}}$): On receiving i-th request, compute $(vk_i, sk_i) \leftarrow \mathsf{SSIG.Key}(\Sigma)$, and return vk_i. Record vk_i to Q_K.
- (Signing Oracle $\mathcal{O}_{\mathsf{sign}}$): On input (vk_i, m), return $\perp$ if vk_i is not recorded. Otherwise, compute $\sigma \leftarrow \mathsf{SSIG.Sign}(\Sigma, sk_i, m)$ and return σ. Record m to $Q_m^{vk_i}$.
- (Proof Oracle $\mathcal{O}_{\mathsf{wi}}$): On input (vk_i, m), return $\perp$ if $0 \leftarrow \mathsf{SSIG.Chk}(\Sigma, vk_i)$ or $m \notin \mathcal{M}_{vk_i}$. Otherwise, compute $\sigma \leftarrow \mathsf{SSIG.Sim}(\Sigma, vk_i, m, \tau)$, and $\pi \leftarrow \mathsf{NIWI.Prf}((\Sigma, vk_i, m), \sigma)$. Then return π.

Definition 5 (Unforgeability with WI-Simulation). A signature scheme in the CRS model is unforgeable against adaptive chosen message and random verification key attacks with witness-indistinguishable simulation if, for any polynomial-time adversary $\mathcal{A}$, the following experiment returns 1 with negligible probability.

Experiment :

$(\Sigma, \tau) \leftarrow \mathsf{SSIG.Crs}(1^\lambda)$

$(m^\star, \sigma^\star, vk^\star) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{sign}}, \mathcal{O}_{\mathsf{vk}}, \mathcal{O}_{\mathsf{wi}}}(\Sigma)$

Return 1 if $vk^\star \in Q_K$ and $m \notin Q_m^{vk^\star}$ and $1 \leftarrow \mathsf{SSIG.Vrf}(\Sigma, vk^\star, m^\star, \sigma^\star)$.

Return 0, otherwise.

6.3 Construction

Let $\Lambda = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, \tilde{g}) \in \{\Lambda_{\mathsf{sym}}, \Lambda_{\mathsf{xdh}}, \Lambda_{\mathsf{sxdh}}\}$ be implicitly given to the algorithms below.

- $\mathsf{SSIG.Crs}(1^\lambda)$: Choose random generators g_z, h_z, g_r, h_u from $\mathbb{G}_1^*$. For $i = 1, \ldots, k$, choose $\chi_i, \gamma_i, \delta_i$ from $\mathbb{Z}_p$, and compute $g_i = g_z^{\chi_i} g_r^{\gamma_i}$ and $h_i = h_z^{\chi_i} h_u^{\delta_i}$. The CRS is set to $\Sigma = (g_z, h_z, g_r, h_u, \{g_i, h_i\}_{i=1}^k)$, and the trapdoor is $tk = (\chi_1, \gamma_1, \delta_1, \ldots, \chi_k, \gamma_k, \delta_k)$.
- $\mathsf{SSIG.Key}(\Sigma)$: Choose $\alpha, \beta \leftarrow \mathbb{Z}_p$ and compute $\{a_i, \tilde{a}_i\}_{i=0}^k \leftarrow \mathsf{Extend}(g_r, \tilde{g}^\alpha)$ and $\{b_i, \tilde{b}_i\}_{i=0}^k \leftarrow \mathsf{Extend}(h_u, \tilde{g}^\beta)$. Let $sk = (\alpha, \beta)$. For some default message $\vec{m}^* \in \mathbb{G}_2^k$, compute a reference signature $\sigma^* = \mathsf{SSIG.Sign}(\Sigma, sk, \vec{m}^*)$ as shown below. Let $vk = (\{a_i, \tilde{a}_i, b_i, \tilde{b}_i\}_{i=0}^k, \sigma^*)$. Output (vk, sk).
- $\mathsf{SSIG.Sign}(\Sigma, sk, \vec{m})$: For $i = 1$ to k and randomly chosen $\zeta_i, \rho_i, \varphi_i \leftarrow \mathbb{Z}_p$, set

$$\begin{aligned}
&(\text{If } m_i \neq 1): \quad s'_i = g_z^{\zeta_i} g_r^{\rho_i} g_i^{-1}, \quad t'_i = m_i, \quad v'_i = h_z^{\zeta_i} h_u^{\varphi_i} h_i^{-1}, \quad w'_i = m_i,\\
&(\text{If } m_i = 1): \quad s'_i = g_z^{\zeta_i} g_r^{\rho_i}, \qquad t'_i = \tilde{g}, \quad v'_i = h_z^{\zeta_i} h_u^{\varphi_i}, \qquad w'_i = \tilde{g}.
\end{aligned}$$

  and

$$z = \prod_{i=1}^k t_i'^{-\zeta_i}, \qquad r = \tilde{g}^\alpha \prod_{i=1}^k t_i'^{-\rho_i}, \qquad u = \tilde{g}^\beta \prod_{i=1}^k w_i'^{-\varphi_i}.$$

Then, compute $\{s_i, t_i\}_{i=1}^k \leftarrow \mathsf{RandSeq}(\{s'_i, t'_i\}_{i=1}^k)$ and $\{v_i, w_i\}_{i=1}^k \leftarrow \mathsf{RandSeq}(\{v'_i, w'_i\}_{i=1}^k)$. Output $\sigma = (z, r, u, \{s_i, t_i, v_i, w_i\}_{i=1}^k)$ as a signature.

- $\mathsf{SSIG.Vrf}(\Sigma, vk, \vec{m}, \sigma)$: Parse σ as $(z, r, u, \{s_i, t_i, v_i, w_i\}_{i=1}^k)$. Output 1 if

$$A = e(g_z, z)\ e(g_r, r) \prod_{i=1}^{k} e(g_i, m_i)\ e(s_i, t_i), \quad \text{and} \tag{14}$$

$$B = e(h_z, z)\ e(h_u, u) \prod_{i=1}^{k} e(h_i, m_i)\ e(v_i, w_i) \tag{15}$$

hold for $A = \prod_{i=0}^{k} e(a_i, \tilde{a}_i)$ and $B = \prod_{i=0}^{k} e(b_i, \tilde{b}_i)$. Output 0, otherwise.

- $\mathsf{SSIG.Chk}(\Sigma, vk)$: Parse vk into $(\{a_i, \tilde{a}_i, b_i, \tilde{b}_i\}_{i=0}^k, \sigma^*)$ and return 0 if it fails. Check if every element of σ^* is in the appropriate group $\mathbb{G}_1$ or $\mathbb{G}_2$, and verify that $1 = \mathsf{SSIG.Vrf}(\Sigma, vk, \vec{m}^*, \sigma^*)$. If any of the checks fails, output 0. Otherwise, output 1.

- $\mathsf{SSIG.Sim}(\Sigma, vk, \vec{m}, tk)$: Take σ^* from vk and parse it into $(z, r, u, \{s_i, t_i, v_i, w_i\}_{i=1}^k)$. By using $tk = (\chi_1, \gamma_1, \delta_1, \ldots, \chi_k, \gamma_k, \delta_k)$, compute (z', r', u') as

$$z' = z \cdot \prod_{i=1}^{k} (m_i/m_i^*)^{-\chi_i}, \quad r' = r \cdot \prod_{i=1}^{k} (m_i/m_i^*)^{-\gamma_i}, \text{ and } \quad u' = u \cdot \prod_{i=1}^{k} (m_i/m_i^*)^{-\delta_i}.$$

Output $\sigma = (z', r', u', \{s_i, t_i, v_i, w_i\}_{i=1}^k)$ as a signature for $\vec{m}$.

6.4 Security

The security of SSIG relies on k-SFP, a generalization of SFP that has k flexible pairings in each relation as formally defined below. In the case of $k = 1$, k-SFP becomes SFP.

Assumption 4 (Simultaneous k-Flexible Pairing Assumption (k-SFP)). Let Λ be a common parameter and let g_z, h_z, g_r, and h_u be random generators of $\mathbb{G}_1$. Let $\{(a_i, \tilde{a}_i), (b_i, \tilde{b}_i)\}_{i=1}^k$ be random elements in $(\mathbb{G}_1 \times \mathbb{G}_2)^{2k}$. For $j = 1, \ldots, q$, let R_j be a tuple $(z, r, u, \{s_i, t_i, v_i, w_i\}_{i=1}^k) \in {\mathbb{G}_2}^3 \times (\mathbb{G}_1 \times \mathbb{G}_2 \times \mathbb{G}_1 \times \mathbb{G}_2)^k$ that satisfies

$$\prod_{i=1}^{k} e(a_i, \tilde{a}_i) = e(g_z, z)\ e(g_r, r) \prod_{i=1}^{k} e(s_i, t_i), \quad \text{and} \tag{16}$$

$$\prod_{i=1}^{k} e(b_i, \tilde{b}_i) = e(h_z, z)\ e(h_u, u) \prod_{i=1}^{k} e(v_i, w_i). \tag{17}$$

Given Λ, g_z, h_z, g_r, h_u, $\{(a_i, \tilde{a}_i), (b_i, \tilde{b}_i)\}_{i=1}^k$, and uniformly chosen $R_1, \ldots, R_q$, it is hard to find $(z^\star, r^\star, u^\star, \{s_i^\star, t_i^\star, v_i^\star, w_i^\star\}_{i=1}^k)$, that fulfill relations (16) and (17). A restriction is that $z^\star \neq 1$ and $z^\star \neq z \in R_j$ for every R_j.

Theorem 6. *For any generic algorithm $\mathcal{A}$, the probability that $\mathcal{A}$ breaks k-SFP with ℓ group operations and pairings is bound by $\mathcal{O}(k^2 \cdot q^2 + \ell^2)/p$.*

Proof of Theorem 6 that justifies the assumption in the generic bilinear group model is in Appendix A.3. As well as Theorem 3, k-SFP implies SDP for any $k \geq 1$. Somewhat contradictory to the fact that k-SFP is a generalization of SFP, we do not see useful reduction between them for $k \geq 2$.

Theorem 7. *Signature scheme* SSIG *is correct and signature-simulatable. It is EUF-CMA with WI-simulation in the multi-user setting if k-SFP holds for Λ.*

Proof. Correctness. Let I (and I^*) denote the set of indexes where $m_i \neq 1$ (and $m_i = 1$, respectively) in $\mathsf{SIG.Sign}$. Regarding the first relation in the verification predicates, we have:

$$\begin{aligned}
&e(g_z, z)\ e(g_r, r)\ \prod_{i=1}^{k} e(g_i, m_i)\ e(s_i, t_i)\ = \\
&= e\left(g_z, \prod_{i=1}^{k} t_i'^{-\zeta_i}\right)\ e\left(g_r, \tilde{g}^{\alpha} \prod_{i=1}^{k} t_i'^{-\rho_i}\right)\ \prod_{i\in I} e\left(g_i, m_i\right)\ e\left(g_z^{\zeta_i} g_r^{\rho_i} g_i^{-1}, t_i'\right)\ \prod_{i\in I^*} e\left(g_z^{\zeta_i} g_r^{\rho_i}, t_i'\right) \\
&= e\left(g_z, \prod_{i=1}^{k} t_i'^{-\zeta_i}\right)\ e\left(g_r, \tilde{g}^{\alpha} \prod_{i=1}^{k} t_i'^{-\rho_i}\right)\ \prod_{i=1}^{k} e\left(g_z^{\zeta_i} g_r^{\rho_i}, t_i'\right) \\
&= e(g_r, \tilde{g}^{\alpha}) = A
\end{aligned}$$

The other relation can be verified in the same manner as $z = \prod_{i=1}^{k} t_i'^{-\zeta_i} = \prod_{i=1}^{k} w_i'^{-\zeta_i}$.

Signature-Simulatability. For every $vk = (\{a_i, \tilde{a}_i, b_i, \tilde{b}_i\}_{i=0}^{k}, \sigma^*)$ such that $1 = \mathsf{SSIG.Chk}(\Sigma, vk)$, every elements in $\{a_i, \tilde{a}_i, b_i, \tilde{b}_i\}_{i=0}^{k}$ is in the correct group $\mathbb{G}_1$ and $\mathbb{G}_2$. Clearly there are (α, β) so that $\prod_{i=0}^{k} e(a_i, \tilde{a}_i) = e(g_r, \tilde{g}^{\alpha})$ and $\prod_{i=0}^{k} e(b_i, \tilde{b}_i) = e(h_u, \tilde{g}^{\beta})$ hold. Therefore such $\{(a_i, \tilde{a}_i, b_i, \tilde{b}_i)\}_{i=0}^{k}$ and (α, β) are a correct key pair. The rest is to show that $\mathsf{SSIG.Sim}$ correctly works to turn valid signature $\sigma^* = (z, r, u, \{s_i, t_i, v_i, w_i\}_{i=1}^{k})$ for $\vec{m}^*$ into a signature $\sigma = (z', r', u', \{s_i, t_i, v_i, w_i\}_{i=1}^{k})$ for message $\vec{m}$. It holds that

$$\begin{aligned}
&e(g_z, z')\, e(g_r, r')\ \prod_{i=1}^{k} e(g_i, m_i)\, e(s_i, t_i) \\
&= e(g_z, z \cdot \prod_{i=1}^{k} (m_i/m_i^*)^{-\chi_i})\, e(g_r, r \cdot \prod_{i=1}^{k} (m_i/m_i^*)^{-\gamma_i})\ \prod_{i=1}^{k} e(g_z^{\chi_i} g_r^{\gamma_i}, m_i)\, e(s_i, t_i) \\
&= e(g_z, z)\, e(g_r, r)\ \prod_{i=1}^{k} e(g_i, m_i^*)\, e(s_i, t_i) = A.
\end{aligned}$$

The other relation $e(h_z, z')\, e(h_u, u')\ \prod_{i=1}^{k} e(h_i, m_i)\, e(v_i, w_i) = B$ can be verified in the same way. Thus the output from $\mathsf{SSIG.Sim}$ is a valid signature for $\vec{m}$.

EUF-CMA with WI-Simulation. Given an instance of k-SFP, we simulate the view of $\mathcal{A}$ in the attack environment as follows.

- (CRS generation) : Do the same as original $\mathsf{SSIG.Crs}$ by using given generators (g_r, h_u, g_z, h_z) in the input instance. The commitment-key is assigned as a CRS, $\Sigma = (g_z, h_z, g_r, h_u, \{g_i, h_i\}_{i=1}^{k})$, and the trapdoor is $tk = (\chi_1, \gamma_1, \delta_1, \ldots, \chi_k, \gamma_k, \delta_k)$.

- (Key Generation Oracle $\mathcal{O}_{\mathsf{vk}}$) : Take $\{a_i, \tilde{a}_i, b_i, \tilde{b}_i\}_{i=1}^k$ from the input instance. Choose $\zeta, \rho, \varphi \leftarrow \mathbb{Z}_p$ and $\tilde{g} \leftarrow \mathbb{G}_2^*$. Then compute $\{a'_i, \tilde{a}'_i\}_{i=0}^k \leftarrow \mathsf{RandSeq}((g_z^{\zeta} g_r^{\rho}, \tilde{g}), (a_1, \tilde{a}_1), \ldots, (a_k, \tilde{a}_k))$ and $\{b'_i, \tilde{b}'_i\}_{i=0}^k \leftarrow \mathsf{RandSeq}((h_z^{\zeta} h_u^{\varphi}, \tilde{g}), (b_1, \tilde{b}_1), \ldots, (b_k, \tilde{b}_k))$. Then simulate a reference signature σ_0 as described below. The verification key is $vk = (\{a'_i, \tilde{a}'_i, b'_i, \tilde{b}'_i\}_{i=0}^k, \sigma_0)$. Record vk to Q_K.
- (Signing Oracle $\mathcal{O}_{\mathsf{sign}}$) : Given message $\vec{m}$ and vk, return $\perp$ if vk is not in Q_K. Take a new tuple $R_j = (z_j, r_j, u_j, \{s_{ij}, t_{ij}, v_{ij}, w_{ij}\}_{i=1}^k)$ from the given instance. Then compute

$$z'_j = z_j \; \tilde{g}^{\zeta} \prod_{i=1}^{k} m_i^{-\chi_i}, \quad r'_j = r_j \; \tilde{g}^{\rho} \prod_{i=1}^{k} m_i^{-\gamma_i}, \quad u'_j = u_j \; \tilde{g}^{\varphi} \prod_{i=1}^{k} m_i^{-\delta_i}. \tag{18}$$

by using (ζ, ρ, φ) used for generating vk in $\mathcal{O}_{\mathsf{vk}}$. The signature is $\sigma_j = (z'_j, r'_j, u'_j, \{s_{ij}, t_{ij}, v_{ij}, w_{ij}\}_{i=1}^k)$.
- (Simulation Oracle $\mathcal{O}_{\mathsf{wi}}$) : Given $\vec{m}$ and vk, return $\perp$ if $0 \leftarrow \mathsf{SSIG.Chk}(\Sigma, vk_i)$ or $m \notin \mathcal{M}_{vk_i}$. If vk is in Q_K, compute $\sigma \leftarrow \mathcal{O}_{\mathsf{sign}}(\vec{m}, vk)$. Otherwise, compute $\sigma \leftarrow \mathsf{SSIG.Sim}(\Sigma, vk, \vec{m}, tk)$. Then compute $\pi \leftarrow \mathsf{NIWI.Prf}((\Sigma, vk, m), \sigma)$ and return π.

When $\mathcal{A}$ outputs $(\vec{m}^{\dagger}, z^{\dagger}, r^{\dagger}, u^{\dagger}, \{s_i^{\dagger}, v_i^{\dagger}, t_i^{\dagger}, w_i^{\dagger}\}_{i=1}^k)$, compute

$$z^{\star} = \left(z^{\dagger}\right) \tilde{g}^{-\zeta} \prod_{i=1}^{k} \left(m_i^{\dagger}\right)^{\chi_i}, \quad r^{\star} = \left(r^{\dagger}\right) \tilde{g}^{-\rho} \prod_{i=1}^{k} \left(m_i^{\dagger}\right)^{\gamma_i}, \quad u^{\star} = \left(u^{\dagger}\right) \tilde{g}^{-\varphi} \prod_{i=1}^{k} \left(m_i^{\dagger}\right)^{\delta_i}, \tag{19}$$

and set $s_i^{\star} = s_i^{\dagger}$, $t_i^{\star} = t_i^{\dagger}$, $v_i^{\star} = v_i^{\dagger}$, and $w_i^{\star} = w_i^{\dagger}$ for $i = 1, \ldots, k$. The reduction algorithm outputs a tuple $(z^{\star}, r^{\star}, u^{\star}, \{s_i^{\star}, t_i^{\star}, v_i^{\star}, w_i^{\star}\}_{i=1}^k)$ and terminates.

It can be verified by inspection that the CRS, the verification-key and the signatures perfectly follow the legitimate distribution. When $\mathcal{A}$ is successful, for the outputs of the reduction algorithm, it holds that

$$\begin{aligned} e(g_z, z^{\star}) \; e(g_r, r^{\star}) \; \prod_{i=1}^{k} e(s_i^{\star}, t_i^{\star}) &= e\left(g_z, \left(z^{\dagger}\right) \tilde{g}^{-\zeta} \prod_{i=1}^{k} \left(m_i^{\dagger}\right)^{\chi_i}\right) \; e\left(g_r, \left(r^{\dagger}\right) \tilde{g}^{-\rho} \prod_{i=1}^{k} \left(m_i^{\dagger}\right)^{\gamma_i}\right) \prod_{i=1}^{k} e\left(s_i^{\dagger}, t_i^{\dagger}\right) \\ &= e\left(g_z^{-\zeta} g_r^{-\rho}, \tilde{g}\right) \; e\left(g_z, z^{\dagger}\right) \; e\left(g_r, r^{\dagger}\right) \prod_{i=1}^{k} e\left(g_i, m_i^{\dagger}\right) \; e\left(s_i^{\dagger}, t_i^{\dagger}\right) \\ &= e(a_0, \tilde{a}_0)^{-1} \; \prod_{i=0}^{k} e(a_i, \tilde{a}_i) = \prod_{i=1}^{k} e(a_i, \tilde{a}_i). \end{aligned}$$

One can also verify that $e(g_z, z^{\star}) \; e(h_u, u^{\star}) \; \prod_{i=1}^{k} e(v_i^{\star}, w_i^{\star}) = \prod_{i=1}^{k} e(b_i, \tilde{b}_i)$ holds in the same way.

What remains is to show that $z^{\star}$ is not in $\{1, z_1, \ldots, z_q\}$. Basically the argument is the same as the one in the proof of Theorem 4 in Section 4.3 by using the fact that the parameters $\zeta, \chi_1, \ldots, \chi_k$ are independent from $\mathcal{A}$'s view. For the same argument to hold, we have to show that when those values are also used for simulating $\mathcal{O}_{\mathsf{wi}}$, they remain information theoretically hidden even after π is seen by the adversary. For vk generated by $\mathcal{O}_{\mathsf{vk}}$, simulation is done just by calling the signing oracle, and, as we know, the signature does not reveal any information about the parameters. On the other hand, for vk that is not generated by $\mathcal{O}_{\mathsf{vk}}$, $\mathsf{SSIG.Chk}$ guarantees that there exists a corresponding signing key sk. Since $\mathsf{NIWI.Prf}$ is witness indistinguishable and there exists randomness that is consistent to a valid signature that could have been generated by the signing algorithm using sk, like in the previous case the parameters remain hidden. Thus the claim holds. ∎

7 Signing Mixed-Group Messages in the SXDH Setting

7.1 Overview

By using the idea of signature chaining, we construct a signature scheme whose message space consists of a mixture of $\mathbb{G}_1$ and $\mathbb{G}_2$. We use two signature schemes: $\mathsf{SIG1}$ signing messages from the space $\mathbb{G}_1^{k_1}$ and $\mathsf{SIG2}$ signing messages from the space $\mathbb{G}_2^{k_2}$. A part of a signature from $\mathsf{SIG2}$ is included into the message given to $\mathsf{SIG1}$ as a joint. Surprisingly, the joint can be as minimal as only one group element s in this case.

7.2 Construction

Let $\mathsf{SIG2}$ be the constant-size signature scheme from Section 4, whose message space is $\mathbb{G}_2^{k_2}$. Let $\mathsf{SIG1}$ be a 'dual' scheme obtained by exchanging $\mathbb{G}_1$ and $\mathbb{G}_2$ in the same scheme. Let the message space of $\mathsf{SIG1}$ is $\mathbb{G}_1^{k_1+1}$. (Note that we use the same letters for variables in a signature. Accordingly, z, r, u, t, and w are in the same group as the input message while s and v are in the other group.) By using these signature scheme, we construct signature scheme XSIG whose message space is $\mathbb{G}_1^{k_1} \times \mathbb{G}_2^{k_2}$ as follows. Let $(\vec{m}, \vec{\tilde{m}})$ be a message in $\mathbb{G}_1^{k_1} \times \mathbb{G}_2^{k_2}$. For vector $\vec{m} \in \mathbb{G}_1^{k_1}$ and single element $s \in \mathbb{G}_1$, let $\vec{m}||s$ denote a vector in $\mathbb{G}_1^{k_1+1}$ obtained by appending s to the end of $\vec{m}$. Let $\Lambda = \Lambda_{\mathsf{sxdh}}$ be given to the functions described below.

- $\mathsf{XSIG.Key}(1^\lambda)$: Run $(vk_1, sk_1) \leftarrow \mathsf{SIG1.Key}(1^\lambda)$ and $(vk_2, sk_2) \leftarrow \mathsf{SIG2.Key}(1^\lambda)$. Output $(vk, sk) = ((vk_1, vk_2), (sk_1, sk_2))$.
- $\mathsf{XSIG.Sign}(sk, (\vec{m}, \vec{\tilde{m}}))$: Run $\sigma_2 = (z, r, s, t, u, v, w) \leftarrow \mathsf{SIG2.Sign}(sk_2, \vec{\tilde{m}})$ and $\sigma_1 = (z', r', s', t', u', v', w') \leftarrow \mathsf{SIG1.Sign}(sk_1, \vec{m}||s)$. Output $\sigma = (\sigma_1, \sigma_2)$.
- $\mathsf{XSIG.Vrf}(vk, (\vec{m}, \vec{\tilde{m}}), (\sigma_1, \sigma_2))$: Take $s \in \mathbb{G}_1$ from σ_2. Run $b_2 = \mathsf{SIG2.Vrf}(vk_2, \vec{\tilde{m}}, \sigma_2)$ and $b_1 = \mathsf{SIG1.Vrf}(vk_1, \vec{m}||s, \sigma_1)$. Output 1 if $b_2 = b_1 = 1$. Output 0, otherwise.

7.3 Security

Theorem 8. *If SIG1 and SIG2 are EUF-CMA, so is XSIG.*

Proof. Suppose that there is a successful adversary that launches chosen message attacks and outputs a valid forgery, $((\vec{m}^\dagger, \vec{\tilde{m}}^\dagger), (\sigma_1^\dagger, \sigma_2^\dagger))$. Consider $s^\dagger$ included in $\sigma_2^\dagger$. Observe that $\sigma_1^\dagger$ is a signature for $m^\dagger||s^\dagger$. We then have 3 cases.

Type-I $\vec{m}^\dagger||s^\dagger$ has never been signed by the signing oracle. This case contradicts to the unforgeability of $\mathsf{SIG1}$.

Type-II $\vec{\tilde{m}}^\dagger$ has never been signed by the signing oracle. This case contradicts to the unforgeability of $\mathsf{SIG2}$.

Type-III Both $\vec{m}^\dagger||s^\dagger$ and $\vec{\tilde{m}}^\dagger$ have been signed by the signing oracle in separate queries. This case contradicts to the DBP assumption.

Since the first two forgery cases are trivial, we focus on Type-III. We construct a reduction algorithm that simulates the environment for adversary $\mathcal{A}$ launching an adaptive chosen message attack on XSIG. The simulator only simulates $\mathsf{SIG2}$ and honestly acts with respect to $\mathsf{SIG1}$. We thus describe the simulation only with respect to $\mathsf{SIG2}$. Given an instance of the DBP assumption, (Λ, g_z, g_r), the simulator works as follows:

- (Key Generation): Choose random h_z and h_u from $\mathbb{G}_1^*$. Then, for $i = 1, \ldots, k_2$, set $g_i = g_z^{\chi_i} g_r^{\gamma_i}$ and $h_i = h_z^{\chi_i} h_u^{\delta_i}$ for random χ_i, γ_i, and δ_i in $\mathbb{Z}_p$. Choose α, β from $\mathbb{Z}_p$ and $\tilde{g}$ from $\mathbb{G}_2^*$. Then compute $\{a_i, \tilde{a}_i\}_{i=0}^1 \leftarrow \mathsf{Extend}(g_r, \tilde{g}^\alpha)$ and $\{b_i, \tilde{b}_i\}_{i=0}^1 \leftarrow \mathsf{Extend}(h_u, \tilde{g}^\beta)$. Output $vk_1 = (g_z, h_z, g_r, h_u, \{g_i, h_i\}_{i=1}^{k_2}, \{a_i, \tilde{a}_i, b_i, \tilde{b}_i\}_{i=0}^1)$.
- (Signature Issuing): Given message $\tilde{m} \in \mathbb{G}_2^{k_2}$, choose ζ, ρ, τ, φ, $\omega \leftarrow \mathbb{Z}_p$ and set

$$z = \tilde{g}^\zeta \prod_{i=1}^{k_2} \tilde{m}_i^{-\chi_i}, \quad r = \tilde{g}^{\zeta\rho/\tau+\alpha} \prod_{i=1}^{k_2} \tilde{m}_i^{-\gamma_i}, \quad s = g_z^\tau g_r^\rho, \quad t = \tilde{g}^{-\zeta/\tau}$$

$$u = \tilde{g}^{\zeta\varphi/\omega+\beta} \prod_{i=1}^{k_2} \tilde{m}_i^{-\delta_i}, \quad v = h_z^\omega h_u^\varphi, \quad w = \tilde{g}^{-\zeta/\omega}.$$

  Output $\sigma_2 = (z, r, s, t, u, v, w)$.

To see the correctness of the simulated signatures, observe that

$$\begin{aligned}
&e(g_z, z)\; e(g_r, r)\; e(s, t) \prod_{i=1}^{k_2} e(g_i, m_i) \\
&= e\left(g_z, \tilde{g}^\zeta \prod_{i=1}^{k_2} m_i^{-\chi_i}\right)\; e\left(g_r, \tilde{g}^{\zeta\rho/\tau+\alpha} \prod_{i=1}^{k_2} m_i^{-\gamma_i}\right)\; e\,(s, t)\; \prod_{i=1}^{k_2} e\,(g_z^{\chi_i} g_r^{\gamma_i}, m_i) \\
&= e\left(g_z, \tilde{g}^\zeta\right)\; e\left(g_r, \tilde{g}^{\zeta\rho/\tau+\alpha}\right)\; e\left(g_z^\tau g_r^\rho, \tilde{g}^{-\zeta/\tau}\right) \\
&= e(g_r, \tilde{g}^\alpha) = \prod_{i=0}^{1} e(a_i, \tilde{a}_i)
\end{aligned}$$

holds. The other verification predicate holds in the same way. It is also not hard to inspect that the signatures follow a proper distribution due to the random coins in the simulation.

Let $\sigma^\dagger = (\sigma_1^\dagger, \sigma_2^\dagger)$, where $\sigma_2^\dagger = (z^\dagger, r^\dagger, s^\dagger, t^\dagger, u^\dagger, v^\dagger, w^\dagger)$, be the forged signature for a message $(\vec{m}^\dagger, \vec{\tilde{m}}^\dagger)$. By the forgery type constrains, there exists a signing query with message $(\vec{m}^\dagger, \vec{\tilde{m}})$ such that $\vec{\tilde{m}} \neq \vec{\tilde{m}}^\dagger$ and its signature $\sigma_2 = (z, r, s, t, u, v, w)$ satisfies $s = s^\dagger$. Accordingly, we have

$$e(g_z, z^\dagger)\; e(g_r, r^\dagger)\; e(s^\dagger, t^\dagger) \prod_{i=1}^{k_2} e(g_i, \tilde{m}_i^\dagger) = e(g_z, z)\; e(g_r, r)\; e(s^\dagger, t) \prod_{i=1}^{k_2} e(g_i, \tilde{m}_i). \tag{20}$$

Recall that $s^\dagger = s = g_z^\tau g_r^\rho$. By dividing the left-hand of the above equation by its right-hand, we have

$$\begin{aligned}
1 &= e\left(g_z, \frac{z^\dagger}{z}\right)\; e\left(g_r, \frac{r^\dagger}{r}\right)\; e\left(s^\dagger, \frac{t^\dagger}{t}\right) \prod_{i=1}^{k_2} e\left(g_i, \frac{\tilde{m}_i^\dagger}{\tilde{m}_i}\right) \\
&= e\left(g_z, \frac{z^\dagger}{z} \prod_{i=1}^{k_2} \left(\frac{\tilde{m}_i^\dagger}{\tilde{m}_i}\right)^{\chi_i}\right)\; e\left(g_r, \frac{r^\dagger}{r} \prod_{i=1}^{k_2} \left(\frac{\tilde{m}_i^\dagger}{\tilde{m}_i}\right)^{\gamma_i}\right)\; e\left(g_z^\tau g_r^\rho, \frac{t^\dagger}{t}\right) \\
&= e(g_z, z^\star)\; e(g_r, r^\star),
\end{aligned}$$

where $z^\star = \frac{z^\dagger}{z}(\frac{t^\dagger}{t})^\tau \prod_i (\frac{\tilde{m}_i^\dagger}{\tilde{m}_i})^{\chi_i}$ and $r^\star = \frac{r^\dagger}{r}(\frac{t^\dagger}{t})^\rho \prod_i (\frac{\tilde{m}_i^\dagger}{\tilde{m}_i})^{\gamma_i}$.

Since $\vec{\tilde{m}}^\dagger \neq \vec{\tilde{m}}$, there exists $i^\star$ such that $\tilde{m}^\dagger_{i^\star}/\tilde{m}_{i^\star} \neq 1$. Observe that $\chi_{i^\star}$ is independent of the view of the adversary. Hence the probability that $z^\star = 1$ is negligible. The reduction algorithm outputs $(z^\star, r^\star)$ as a valid answer to the given instance of DBP. ∎

8 Strongly Unforgeable Signatures

The following generic construction of sEUF-CMA signature scheme is in [9]. Let SIG be a signature scheme and OTS be a one-time signature scheme. The construction requires that the message space of SIG covers the public-key space of OTS.

- $\mathsf{FSIG1.Key}(1^\lambda)$: Run $(vk, sk) \leftarrow \mathsf{SIG.Key}(1^\lambda)$. Output (vk, sk).
- $\mathsf{FSIG1.Sign}(sk, \vec{m})$: $(vk_o, sk_o) \leftarrow \mathsf{OTS.Key}(1^\lambda)$, $\sigma_1 \leftarrow \mathsf{SIG.Sign}(sk, vk_o||\vec{m})$, $\sigma_2 \leftarrow \mathsf{OTS.Sign}(sk_o, \sigma_1)$. Output $\sigma = (vk_o, \sigma_1, \sigma_2)$.
- $\mathsf{FSIG1.Vrf}(vk, \vec{m}, \sigma)$: Parse σ into $(vk_o, \sigma_1, \sigma_2)$. Compute $b_1 \leftarrow \mathsf{SIG.Vrf}(vk, vk_o||\vec{m}, \sigma_1)$ and $b_2 \leftarrow \mathsf{OTS.Vrf}(vk_o, \sigma_1, \sigma_2)$. Output 1 if $b_2 = b_1 = 1$. Output 0, otherwise.

As shown in [9], signature scheme FSIG1 is strongly EUF-CMA if SIG is EUF-CMA and OTS is sEUF-CMA against one-time chosen message attacks. In the one-time chosen message attacks, the adversary is allowed to make at most one signing query. We refer to [9] for a proof.

By instantiating SIG and OTS by the ones in Section 4 and Appendix C.1 with setting $\Lambda = \Lambda_{\mathsf{sym}}$, the resulting FSIG1 outputs a signature of 32 group elements ($|vk_o| = 22$, $|\sigma_1| = 7$, $|\sigma_2| = 3$) which is a constant in the size of $\vec{m}$.

We can gain efficiency by using the same bases in SIG and OTS in the above instantiation. Let FSIG2 denote this variant. Concretely, in FSIG2, one-time signature OTS takes bases $(g_z, h_z, g_r, h_u, g_1, h_1, \ldots, g_7, h_7)$ from those of SIG. Then vk_o only includes a and b. As a result, a signature of FSIG2 consists of 12 group elements.

The generic security argument for FSIG1 no longer holds for FSIG2 since SIG and OTS are not independent. We are still able to show that FSIG2 is sEUF-CMA as follows.

Theorem 9. *Signature scheme FSIG2 is sEUF-CMA if SFP holds for $\Lambda = \Lambda_{\mathsf{sym}}$.*

Proof. First observe that we cannot show a black-box reduction to the security of SIG and OTS by using their signing oracles since they share the bases. We instead construct reduction to their underlying assumptions. This is possible because, in both security proofs for SIG and OTS, bases $(g_1, h_1, \ldots, g_7, h_7)$ are set in the same manner with respect to (g_z, h_z, g_r, h_u). Thus, while we simulate the signing oracle for SIG, we can also simulate signatures of OTS.

Let $\mathcal{O}_{\mathsf{sign}}$ be the signing oracle of FSIG2. Suppose that an adversary outputs a valid forgery $(vk_o^\dagger, \sigma_1^\dagger, \sigma_2^\dagger, \vec{m}^\dagger)$. Let $Q_i = (vk_o, \sigma_1, \sigma_2, \vec{m})$ for $i = 1, \ldots, q$ be the record of interaction between the adversary and $\mathcal{O}_{\mathsf{sign}}$.

For a type of adversary that causes $(vk_o^\dagger, \vec{m}^\dagger) \neq (vk_o, \vec{m})$ for any Q_i, we construct a reduction to SFP by simulating SIG as shown in the proof of Theorem 4. We also simulate OTS as shown in the proof of Theorem 15. Note that the simulation of OTS is possible since the way bases g_i and h_i are set in simulating SIG is exactly the same as that in simulating OTS. Thus we can successfully simulate FSIG2 by using these simulated SIG and OTS. It is important to see that exponents hidden in g_i and h_i remain independent of the view of the adversary even with the simulation of OTS. Thus a successful forgery results in a contradiction to SFP as shown in the proof of Theorem 4.

For the other type of adversary that causes $(vk_o^\dagger, \vec{m}^\dagger) = (vk_o, \vec{m})$ and $(\sigma_1^\dagger, \sigma_2^\dagger) \neq (\sigma_1, \sigma_2)$ for some $Q_{i^\star}$, we show a reduction to SDP by simulating OTS as shown in the proof of Theorem 15. Since simulation of SIG needs an instance of SFP, we generate a random instance of SFP from that of SDP as follows. Given an SDP instance (g_z, h_z, g_r, h_u), set $a = g_r^\alpha$ and $b = h_u^\beta$. Then for $j = 1, \ldots, q$, compute reference $R_j = (z, r, u, s, t, v, w)$ by choosing $\zeta \leftarrow \mathbb{Z}_p$ and setting $z = \tilde{g}^\zeta$, $r' = \tilde{g}^\alpha$, $s' = g_z$, $t' = \tilde{g}^\zeta$, $u' = \tilde{g}^\beta$, $v' = h_z$, $w' = \tilde{g}^\zeta$ and applying $(r, s, t, u, v, w) \leftarrow \mathsf{SigRand}(r', s', t', u', v', w')$. The rest of the simulation for SIG is the same as that in the proof of Theorem 4. As well as the previous case, the simulation of SIG retains independent of the exponents hidden in g_i and h_i. Thus a successful forgery contradicts to SDP as shown in the proof of Theorem 15. Finally, applying Theorem 3 to reduce SDP to SFP completes the proof. ∎

9 Applications

In some cryptographic protocols, the existing state of the art constructions achieve the desired security properties with good efficiency, whereas for others certain compromises are made (achieving slightly weaker notion of security or being somewhat inefficient). Below we present constructions for round-optimal blind signatures following the framework of [28], the efficient instantiation of which has been an open problem since Crypto'06; and efficient fully secure group signatures supporting concurrent join procedure, with previous constructions being not in the standard model, secure under weaker model, not supporting concurrent join procedure, or being inefficient. Our signature schemes not only embody known modular protocol designs, but also achieve an excellent efficiency. These are good examples that enlightens the usefulness of modular protocol design and significance of developing efficient building blocks.

9.1 Round-Optimal Blind Signatures

We present an efficient instantiation of Fischlin's round-optimal blind signature scheme [28]. In fact, we use the modification of [40, 3] for which the generic construction uses a non-interactive witness indistinguishable (NIWI) proof system and a simulatable signature scheme. This gives the first efficient round-optimal non-committing blind signature scheme adaptively secure in the universally composability framework [24].

The structure of the framework is the following. A user commits to a message m with opening d and send the commitment c to the signer. The signer signs commitment c and return the signature σ to the user. Then the user computes a NIWI proof π with witness (c, d, σ) for the fact that he knows a commitment c of the message m, he knows the correct opening d, and he has a valid signature on c with respect to a verification key vk of the signer. The security follows from the generic framework in [3].

To instantiate this generic scheme, we use the GS proof system, the simulatable signature scheme SSIG from Section 6 for $k = 1$ (i.e. for signing only a single group element), and the commitment scheme $\mathsf{TC2}$ in Appendix B.2. In fact, any commitment scheme suffices for our purpose as long as commitment key, commitments, and openings to be group elements and the verification is by pairing product equations. The choice of $\mathsf{TC2}$ is due to the efficiency; it has the smallest commitment size. The commitment scheme $\mathsf{TC2}$ could be viewed as a "pairing-based variant" of Pedersen commitment [48], and, indeed, is almost as efficient.

Let $\Lambda \in \{\Lambda_{\mathsf{sym}}, \Lambda_{\mathsf{sxdh}}\}$ be the common parameter. Let $(\Sigma_{\mathsf{com}}, tk) \leftarrow \mathsf{TC2.Key}(1^\lambda)$, $(\Sigma_{\mathsf{sig}}, tk') \leftarrow \mathsf{SSIG.Crs}(1^\lambda)$. Let Σ_{niwi} be the common reference string for the GS proof system in the simulation mode. Concretely, those are $\Sigma_{\mathsf{com}} = f \in \mathbb{G}_2$, $\Sigma_{\mathsf{sig}} = (g_z, h_z, g_r, h_u, g_1, h_1) \in \mathbb{G}_1^6$, and Σ_{niwi} set up in

Scheme	#(rounds)	Communication	Signature Size	Security Model	Assumption
Oka06[46]	4	$3^{[N^2]} + 4^{[1]} + 10^{[p]}$	$4^{[1]} + 1^{[p]}$	SA	2SDH,DCR
KZ08[44]	6	$9^{[N^2]} + 7^{[1]} + 7^{[p]}$	$4^{[1]} + 1^{[p]}$	UC	2SDH,DCR
Fuc10[29]	2	$22^{[1]}$	$30^{[1]}$	SA	DAHSDH,HDL,DLIN
(ours)	2	$8^{[1]}$	$28^{[1]}$	UC	SFP, DLIN

Table 1: Summary of efficiency of concurrently secure efficient blind signatures. Columns for "Communication" and "Signature Size" count the number of elements, indicating the groups they belong to ($[N^2]$, $[1]$, and $[p]$, respectively, for $\mathbb{Z}_{N^2}$, $\mathbb{G}_1$ and $\mathbb{Z}_p$). UC: Universally Composable Security with Adaptive Corruption [25, 3]. SA: Stand-Alone Security. 2SDH: 2-Variable Strong Diffie-Hellman Assumption [46]. DCR: Decision Composite Residuosity [47]. DAHSDH,HDL: See [29]

the way the simulated CRS is created according to Section 2.4. The CRS for the blind signature scheme is $\Sigma = (\Lambda, \Sigma_{\mathsf{com}}, \Sigma_{\mathsf{sig}}, \Sigma_{\mathsf{niwi}})$. A signer runs $(vk, sk) \leftarrow \mathsf{SSIG.Key}(\Sigma_{\mathsf{sig}})$ where $vk = (A, B, \sigma^*)$ and publish vk as his verification key. The blind signature issuing protocols is as follows:

- On input $m \in \mathbb{Z}_p$, a user computes $(c, d) \leftarrow \mathsf{TC2.Com}(\Sigma_{\mathsf{com}}, m)$ where $(c, d) = (\tilde{g}^m f^\delta, g^\delta) \in \mathbb{G}_2 \times \mathbb{G}_1$. Then the user sends c to the signer.
- The signer computes $(z, r, s, t, u, v, w) \leftarrow \mathsf{SSIG.Sign}(sk, c)$ and sends σ to the user.
- The user computes $(r', s', t', u', v', w') \leftarrow \mathsf{SigRand}(r, s, t, u, v, w)$ as in Section 4.4, and gives a GS-proof π with a witness (c, d, z, r', u') for pairing product equations

$$e(g, c)\, e(d, f^{-1}) = e(g, \tilde{g}^m), \tag{21}$$

$$e(g_z, z)\, e(g_r, r')\, e(g_1, c) = A \cdot e(s', t')^{-1}, \tag{22}$$

$$e(h_z, z)\, e(h_u, u')\, e(h_1, c) = B \cdot e(v', w')^{-1}. \tag{23}$$

  Then output a signature $\sigma = (s', t', v', w', \pi)$ for m.

Given (σ, m), a verifier accepts $\sigma = (s', t', v', w', \pi)$ if π is a correct GS-proof with respect to relations (21), (22), and (23).

In the construction, the use of $\mathsf{SigRand}$ is for better efficiency and does not affect to the framework due to the nature of perfect randomness. The resulting blind signature consists of 4 group elements, 5 GS commitments to group elements, and proof elements for 3 pairing product equations. Note that when $\Lambda = \Lambda_{\mathsf{sym}}$, we could swap the elements in the second pairing of the first equation and get all three equations to be one-sided pairing products. Thus, the size of final blind signature is 28 group elements for $\Lambda = \Lambda_{\mathsf{sym}}$ using GS-proof system with DLIN setting. It can be reduced to 26 group elements (precisely 8 in $\mathbb{G}_1$ and 18 in $\mathbb{G}_2$) for $\Lambda = \Lambda_{\mathsf{sxdh}}$ using GS-proof system with SXDH setting. The communication complexity is quite low. Only 8 group elements are exchanged in total, and achieves optimal 2 moves. These figures could be a good efficiency standard for "crafted" constructions to compare.

By replacing SSIG with SIG from Section 4, one could also instantiate the very original Fischlin's scheme that is secure against static adversaries. This, however, requires NIZK proofs and hence becomes less efficient; NIZK requires that we replace A and B with their pairing product representations as originally described for SIG in Section 4. We also remark that the construction can be extended to a *partially-blind* scheme [2] as SSIG (and SIG) can sign multiple group elements at once.

Table 1 summaries efficiency of some known blind signature schemes. There are other schemes that achieve concurrent security without random oracles, e.g., [21, 43, 40, 44]. [46] is a representative from those without GS-proofs. Sizes for [46] vary in parameter setting and include some approximation. Numbers for [44] translates numbers in $\mathbb{Z}_{N^3}$ and $\mathbb{Z}_N$ into that of $\mathbb{Z}_{N^2}$ with appropriate factors . (Precisely, $9^{[N^2]}$ is a translation of $1^{[N^3]} + 6^{[N^2]} + 3^{[N]}$.) Our instantiation is very strong in communication while the schemes in [46, 44] with classical blind-then-unblind structure have an advantage in the signature size.

9.2 Group Signatures with Concurrent Join

This section highlights a useful property of our signature schemes that the message space is compatible with the verification key space. In particular, we present the most efficient instantiation of a group signature scheme that allows efficient concurrent join protocol [42].

In the symmetric setting $\Lambda = \Lambda_{\mathsf{sym}}$, the message space of $\mathsf{USIG1}$ from Section 5 includes the verification key space. This allows Alice to sign Bob's key and Bob can sign Charlie's key and so on. Such a chaining can be hidden by applying NIZK. A signature scheme that allows to sign its own verification key is introduced as automorphic signatures in [29]. It is proven to have some interesting high-level applications such as proxy signatures.

Conceptually, a group signature scheme is a special case of such anonymous delegation system with only one hop of delegation. As sketched in [23] and embodied in [42], the above single-round certification protocol between Alice and Bob brings some favorable properties in the construction of efficient group signature schemes. In the following, we revisit the general idea of [23, 42, 36] with CPA-anonymity [12] by using terminology of proof of knowledge. The construction extends to CCA-anonymity by following the generic construction in [36]. Let $\mathsf{SIG0}$ and $\mathsf{SIG1}$ be signature schemes, and NIWI be a witness indistinguishable proof of knowledge system. A group signature, GSIG, consists of 5 algorithms $\{\mathsf{Setup}, \mathsf{Join}, \mathsf{Sign}, \mathsf{Vrf}, \mathsf{Open}\}$ such that:

- $\mathsf{GSIG.Setup}$ is a setup algorithm that takes security parameter 1^λ and runs $(vk_{\rm c}, sk_{\rm c}) \leftarrow \mathsf{SIG0.Key}(1^\lambda)$ and also sets up a CRS Σ_{niwi} and a trapdoor $sk_{\rm o}$ for NIWI. Group verification-key is $vk_{\rm g} = (vk_{\rm c}, \Sigma_{\mathsf{niwi}})$. The certification-key $sk_{\rm c}$ is given privately to the issuer and the opening-key $sk_{\rm o}$ is given privately to the opener.

- $\mathsf{GSIG.Join}$ is an interactive protocol between a group member and the issuer. The group member generates his own key-pair $(vk_{\rm u}, sk_{\rm u}) \leftarrow \mathsf{SIG1.Key}(1^\lambda)$ and send $vk_{\rm u}$ to the issuer. The issuer signs on $vk_{\rm u}$ by $\sigma_{\rm c} \leftarrow \mathsf{SIG0.Sign}(sk_{\rm c}, vk_{\rm u})$ and send the certificate $\sigma_{\rm c}$ to the member.

- $\mathsf{GSIG.Sign}$ is a signing algorithm run by a group member to sign message m. It consists of signing on message m by $\sigma_{\rm u} \leftarrow \mathsf{SIG1.Sign}(sk_{\rm u}, m)$ and generating a non-interactive witness indistinguishable proof of knowledge $\pi \leftarrow \mathsf{NIWI.Prf}(\Sigma_{\mathsf{niwi}}, \mathrm{pub}, \mathrm{wit})$ that proves relation $1 = \mathsf{SIG0.Vrf}(vk_{\rm c}, vk_{\rm u}, \sigma_{\rm c})$ and $1 = \mathsf{SIG1.Vrf}(vk_{\rm u}, m, \sigma_{\rm u})$ with respect to witness $\mathrm{wit} = (vk_{\rm u}, \sigma_{\rm c}, \sigma_{\rm u})$ and public information $\mathrm{pub} = (vk_{\rm c}, m)$. Final output is π, which is a group signature.

- $\mathsf{GSIG.Vrf}$ takes $(vk_{\rm g}, m, \pi)$ as input and verifies correctness of π by verifying π as a NIWI proof with respect to $\mathrm{pub} = (vk_{\rm c}, m)$ and CRS Σ_{niwi}.

- $\mathsf{GSIG.Open}$ is an opening algorithm run by the opener who has opening-key $sk_{\rm o}$. Given π and $sk_{\rm o}$ as input, the algorithm runs the knowledge extractor of the NIWI proof system and extracts witness $(vk_{\rm u}, \sigma_{\rm c}, \sigma_{\rm u})$. The exposed verification key $vk_{\rm u}$ identifies the group member who actually created π. This algorithm will be associated with another algorithm that publicly verifies the correctness of the opening.

Theorem 10. *Group signature scheme* GSIG *is CPA-anonymous, traceable, and non-frameable.*

We refer to [12] and [8] for formal definitions of the security notions stated in the theorem. Intuitively, CPA-anonymity is that the adversary cannot distinguish group signatures from two members. As CPA security, the adversary is not given oracle access to the opener. Traceability guarantees that once a group signature is opened, it identifies a group member who once completed $\mathsf{GSIG.Join}$. Non-frameability means that no one but a group member can issue a valid group signature that points to the member if opened.

Proof. CPA-anonymity follows directly from the (computational) WI property [39] of the proof system NIWI. For traceability, suppose that there is a valid signature π on message m. Due to the knowledge soundness of NIWI, the opener can extract $(vk_{\mathrm{u}}, \sigma_{\mathrm{c}}, \sigma_{\mathrm{u}})$ from π and $(vk_{\mathrm{u}}, \sigma_{\mathrm{c}})$ fulfills $1 = \mathsf{SIG0.Vrf}(vk_{\mathrm{c}}, vk_{\mathrm{u}}, \sigma_{\mathrm{c}})$. If vk_{u} does not point any group member registered through $\mathsf{GSIG.Join}$, σ_{c} is a valid forgery for $\mathsf{SIG0}$, which contradicts to the EUF-CMA property of $\mathsf{SIG0}$. Thus vk_{u} allows tracing. For non-frameability, suppose that the opener extracts $(vk_{\mathrm{u}}, \sigma_{\mathrm{c}}, \sigma_{\mathrm{u}})$ from a group signature on message m. If $1 = \mathsf{SIG1.Vrf}(vk_{\mathrm{u}}, m, \sigma_{\mathrm{u}})$ holds but the owner of vk_{u} have never signed on m, it is a valid forgery against $\mathsf{SIG1}$, contradicting the EUF-CMA property of $\mathsf{SIG1}$. ∎

As mentioned in [42], the above framework has been known without efficient instantiation in the standard model. Using our main signature scheme SIG as $\mathsf{SIG0}$ and GS-proof system as NIWI, we can instantiate the construction with efficiency. We assess the efficiency in the setting $\Lambda = \Lambda_{\mathsf{sym}}$ as follows. Let $\mathsf{SIG1}$ be a signature scheme whose verification key vk_{u} and signature σ_{u} consist of α and β group elements, respectively. Let γ be the number of group elements needed to prove relation $1 = \mathsf{SIG1.Vrf}(vk_{\mathrm{u}}, m, \sigma_{\mathrm{u}})$ including GS-commitments for vk_{u} and σ_{u}. Regardless of size vk_{u} to be certified, our $\mathsf{SIG0}$ outputs σ_{c} of size 7. Since 4 out of the 7 elements in σ_{c} can be perfectly randomized and given in the clear as we have done in Section 9.1, we need only 3 GS-commitments in proving relation $1 = \mathsf{SIG0.Vrf}(vk_{\mathrm{c}}, vk_{\mathrm{u}}, \sigma_{\mathrm{c}})$, which consists of two one-sided pairing product equations and costs 6 elements in a proof. (Commitments of vk_{u} is already included in γ.) In total we have (Group Sig Size) $= 19 + \gamma$. One can instantiate $\mathsf{SIG0}$ with the signature scheme in [26], that has $9\alpha + 4$ elements in σ_{c} and $3\alpha + 3$ one-sided and 3α double-sided pairing product equations in $\mathsf{SIG0.Vrf}$. In that case, the size of a group signature is (Group Sig Size) $= 63\alpha + 21 + \gamma$.

If we instantiate $\mathsf{SIG1}$ with full EUF-CMA Boneh-Boyen signature scheme from [10], vk_{u} consists of $\alpha = 4$ group elements (including the bases). A signature consists of one group element and one scalar value but the scalar value is totally random and independent of the verification-key. So we have $4+1$ GS-commitments in proving $1 = \mathsf{SIG1.Vrf}(vk_{\mathrm{u}}, m, \sigma_{\mathrm{u}})$. The verification predicate consists of a double-sided pairing product equation, which yields 9 group elements in a proof. In total, we have $\gamma = 24$ and a group signature consists of 43 group elements and 1 scalar value. With [26] for $\mathsf{SIG0}$, the signature size will be 297 group elements and 1 scalar value. These figures can be slightly decreased by using the GS proofs in the SXDH setting.

Table 2 summarizes some efficient group signature schemes that provide CPA-anonymity in the standard model with non-interactive assumptions. ([42] allows concurrent join but the security is argued in the random oracle model [7]. A scheme in [4] is non-frameable but only allows sequential join. It bases on strong interactive assumptions.) [18] (and also [17]) provides efficiency with reasonable assumptions but are frameable. The scheme in [36] is non-frameable but does not allow concurrent join as their Join protocol includes 6 rounds of interaction. Also, the traceability of [36] demands a strong dedicated assumption on top of the security of the building blocks. Our construction $\mathsf{GSIG}(\mathsf{SIG}+\mathrm{BB}[10])$ yields a signature that includes 15 more group elements than that of [36]. This is the price for achieving concurrent join property and allowing very simple and modular security argument without dedicated assumptions.

Scheme	Concurrent Join	Non-Frameability	Signature Size	Assumptions
BW07[18]	yes	no	$6^{[N]}$	SD, HSDH
Gro07[36]	no	yes	$28^{[1]}$	SDH, q-U, DLIN
GSIG([26]+BB[10])	yes	yes	$297^{[1]} + 1^{[p]}$	DLIN,SDH,HSDH
GSIG(SIG+BB[10])	yes	yes	$43^{[1]} + 1^{[p]}$	SFP, SDH

Table 2: Summary of efficiency and properties of group signature schemes with CPA-anonymity. The signature size counts the number of elements and indicating the groups they belong to ([1], $[N]$, and $[p]$ respectively for $\mathbb{G}_1$, $\mathbb{Z}_N$, and $\mathbb{Z}_p$). SD: Subgroup Decision Assumption [14]. q-U: See [36].

Some final remarks follow:

- CCA-anonymity is obtained by following the approach in [36], which uses a strong one-time signature scheme and a selective-tag CCA secure tag-based public-key encryption scheme. By using the same instantiation as in [36], this strengthening costs extra 15 group elements in a signature. Accordingly, we have a CCA-anonymous group signature scheme with concurrent join whose signature consists of 58 group elements and one scalar value.

- One of the advantages of using our SIG for $\mathsf{SIG0}$ is that it allows to insert a warranty in the clear to σ_{c} so that the signing policy given to a group member is explicit. Due to the constant-size property of SIG, this useful extension can be done without impacting to the size of the group signature (except for the warranty itself) at all.

10 Conclusion

This paper presented a practical signature scheme all components of which are group elements in bilinear settings. Signing arbitrary k group elements at the same time results in a signature of size only 7 group elements. This solves an open problem in [35] (explicitly stated in [26]). Its technically interesting properties are enlightened by presenting variations with advanced properties. Combined with Groth-Sahai proof system, our signature schemes give handy and reasonably practical solutions to many cryptographic tasks.

The most challenging open problem is to base security on weaker and well studied assumptions while retaining the efficiency and compatibility. Also, it is left as an open problem to construct a homomorphic trapdoor commitment scheme for group elements with constant-size commitments in the base groups.

References

[1] M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M. Ohkubo. Structure-preserving signatures and commitments to group elements. In *Advances in Cryptology - CRYPTO 2010*, LNCS, pages 209–237. Springer-Verlag, 2010. (Cited on page 1, 2, 36.)

[2] M. Abe and E. Fujisaki. How to date blind signatures. In K. Kim and T. Matsumoto, editors, *Advances in Cryptology – ASIACRYPT '96*, volume 1163 of *LNCS*, pages 244–251. Springer-Verlag, 1996. (Cited on page 25.)

[3] M. Abe and M. Ohkubo. A framework for universally composable non-committing blind signatures. In M. Matsui, editor, *Advances in Cryptology - ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 435–450. Springer-Verlag, 2009. (Cited on page 2, 3, 15, 24, 25.)

[4] G. Ateniese, J. Camenisch, S. hohenberger, and B. de Medeiros. Practical group signatures without random oracles. Cryptology ePrint Archive, Report 2005/385, 2005. `http://eprint.iacr.org`. (Cited on page 27.)

[5] M. Belenkiy, M. Chase, M. Kohlweiss, and A. Lysyanskaya. Non-interactive anonymous credentials. In R. Canetti, editor, *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008*, volume 4948 of *LNCS*. Springer-Verlag, 2008. Also available on IACR ePrint Archive, 2007/384. (Cited on page 2.)

[6] M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements and a construction based on general assumptions. In E. Biham, editor, *Advances in Cryptology - EUROCRPYT 2003*, volume 2656 of *LNCS*, pages 614–629. Springer-Verlag, 2003. (Cited on page 2.)

[7] M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *First ACM Conference on Computer and Communication Security*, pages 62–73. Association for Computing Machinery, 1993. (Cited on page 27.)

[8] M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In A. Menezes, editor, *Topics in Cryptology – CT-RSA 2005*, volume 3376 of *LNCS*, pages 136–154. Springer-Verlag, 2005. Full version available at IACR e-print 2004/077. (Cited on page 2, 27.)

[9] M. Bellare and S. Shoup. Two-tier signatures, strongly unforgeable signatures, and fiat-shamir without random oracles. In *Proceedings of the 10th International Conference on Theory and Practice of Public-Key Cryptography - PKC 2007*, volume 4450 of *LNCS*, pages 201–216. Springer-Verlag, 2007. (Cited on page 23.)

[10] D. Boneh and X. Boyen. Short signatures without random oracles. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology — Eurocrypt '04*, volume 3027 of *LNCS*, pages 56–73. Springer-Verlag, 2004. (Cited on page 27, 28.)

[11] D. Boneh and X. Boyen. Short signatures without random oracles and the sdh assumption in bilinear groups. *Journal of Cryptology*, 21(2):149–177, 2008. (Cited on page 2, 3.)

[12] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In M. Franklin, editor, *Advances in Cryptology — CRYPTO '04*, volume 3152 of *LNCS*, pages 41–55. Springer-Verlag, 2004. (Cited on page 2, 4, 26, 27.)

[13] D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In E. Biham, editor, *Advances in Cryptology - EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 416–432. Springer-Verlag, 2003. (Cited on page 2.)

[14] D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-DNF formulas on ciphertexts. In J. Kilian, editor, *Theory of Cryptography Conference– TCC'2005*, volume 3378 of *LNCS*, pages 325–341. Springer-Verlag, 2005. (Cited on page 28.)

[15] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. In C. Boyd, editor, *Advances in Cryptology – Asiacrypt 2001*, volume 2248 of *LNCS*, pages 514–532. Springer-Verlag, 2001. (Cited on page 39.)

[16] X. Boyen. The uber-assumption family: A unified complexity framework for bilinear groups. In *Pairing 2008*, volume 5209 of *LNCS*, pages 39–56. Springer-Verlag, 2010. (Cited on page 6.)

[17] X. Boyen and B. Waters. Compact group signatures without random oracles. In *Advances in Cryptology — Eurocrypt '06*, volume 4004 of *LNCS*, pages 427–444. Springer-Verlag, 2006. Full version available from IACR ePrint Archive 2005/381. (Cited on page 27.)

[18] X. Boyen and B. Waters. Full-domain subgroup hiding and constant-size group signatures. In *Public Key Cryptography—PKC 2007*, volume 4450 of *LNCS*, pages 1–15. Springer-Verlag, 2007. Available at `http://www.cs.stanford.edu/~xb/pkc07/`. (Cited on page 27, 28.)

[19] J. Camenisch, N. Chandran, and V. Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In *Advances in Cryptology - EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 351–368. Springer-Verlag, 2009. (Cited on page 2.)

[20] J. Camenisch, M. Kohlweiss, and C. Soriente. An accumulator based on bilinear maps and efficient revocation for anonymous credentials. In *Public Key Cryptography, PKC 2009*, volume 5443 of *LNCS*, pages 481–500. Springer-Verlag, 2009. (Cited on page 2.)

[21] J. Camenisch, M. Koprowski, and B. Warinschi. Efficient blind signatures without random oracles. In C. Blundo and S. Cimato, editors, *Security in Communication Networks, 4th International Conference, SCN 2004, Amalfi, Italy, September 8-10, 2004, Revised Selected Papers*, volume 3352 of *LNCS*, pages 134–148. Springer-Verlag, 2005. (Cited on page 26.)

[22] J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *Advances in Cryptology — CRYPTO '04*, volume 3152 of *LNCS*, pages 56–72. Springer-Verlag, 2004. (Cited on page 2.)

[23] J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. In B. S. Kaliski Jr., editor, *Advances in Cryptology — CRYPTO '97*, volume 1294 of *LNCS*, pages 410–424. Springer-Verlag, 1997. (Cited on page 26.)

[24] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings of the 42nd IEEE Annual Symposium on Foundations of Computer Science*, pages 136–145, 2001. (Cited on page 24.)

[25] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. Technical Report 2000/067, IACR e-print Archive, 2005. 2nd version updated on 13 Dec 2005. (Cited on page 25.)

[26] J. Cathalo, B. Libert, and M. Yung. Group encryption: Non-interactive realization in the standard model. In M. Matsui, editor, *Advances in Cryptology - ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 179–196. Springer-Verlag, 2009. (Cited on page 2, 5, 27, 28, 36, 37.)

[27] J. H. Cheon. Security analysis of hte strong diffie-hellman problem. In *Advances in Cryptology — Eurocrypt '06*, volume 4004 of *LNCS*, pages 1–11. Springer-Verlag, 2006. (Cited on page 3.)

[28] M. Fischlin. Round-optimal composable blind signatures in the common reference model. In C. Dwork, editor, *Advances in Cryptology — CRYPTO '06*, volume 4117 of *LNCS*, pages 60–77. Springer-Verlag, 2006. (Cited on page 2, 3, 15, 24.)

[29] G. Fuchsbauer. Automorphic signatures in bilinear groups. IACR ePrint Archive 2009/320. Version 20091217:153918 (dated Dec. 17, 2009)., 2009. (Cited on page 1, 2, 3, 25, 26.)

[30] G. Fuchsbauer, D. Pointcheval, and D. Vergnaud. Transferable anonymous constant-size fair e-cash. IACR ePrint Archive 2009/146. Also to appear in CANS 2009., 2009. (Cited on page 2.)

[31] S. Galbraith, K. Paterson, and N. Smart. Pairings for cryptographers. Technical Report 2006/165, IACR ePrint archive, 2006. (Cited on page 4.)

[32] S. D. Galbraith and V. Rotger. Easy decision-diffie-hellman groups. *LMS Journal of Computation and Mathematics*, 7:2004, 2004. (Cited on page 4.)

[33] S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, April 1988. (Cited on page 4.)

[34] M. Green and S. Hohenberger. Universally composable adaptive oblivious transfer. In J. Pieprzyk, editor, *Advances in Cryptology - ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 179–197. Springer-Verlag, 2008. Preliminary version: IACR ePrint Archive 2008/163. (Cited on page 2.)

[35] J. Groth. Simulation-sound nizk proofs for a practical language and constant size group signatures. In X. Lai and K. Chen, editors, *Advances in Cryptology - ASIACRYPT 2006*, volume 4284 of *LNCS*, pages 444–459. Springer-Verlag, 2006. (Cited on page 2, 28.)

[36] J. Groth. Fully anonymous group signatures without random oracles. In *Advances in Cryptology – Asiacrypt'07*, volume 4833 of *LNCS*, pages 164–180. Springer-Verlag, 2007. (Cited on page 26, 27, 28.)

[37] J. Groth. Homomorphic trapdoor commitments to group elements. Cryptology ePrint Archive, Report 2009/007, January 2009. (Cited on page 1, 5, 36, 37.)

[38] J. Groth. Homomorphic trapdoor commitments to group elements. Unpublished Manuscript, 2010. (Cited on page 5, 36.)

[39] J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Advances in Cryptology — Eurocrypt '08*, volume 4965 of *LNCS*, pages 415–432. Springer-Verlag, 2008. Full version available: IACR ePrint Archive 2007/155. (Cited on page 2, 6, 27.)

[40] C. Hazay, J. Katz, C. Koo, and Y. Lindell. Concurrently-secure blind signatures without random oracles or setup assumptions. In *Theory of Cryptography Conference, TCC 2007*, volume 4392 of *LNCS*, pages 323–341. Springer-Verlag, 2007. (Cited on page 15, 24, 26.)

[41] A. Kiayias and M. Yung. Group signatures with efficient concurrent join. In *Advances in Cryptology – Eurocrypt 2005*, volume 3494 of *LNCS*, pages 198–214. Springer-Verlag, 2005. (Cited on page 2.)

[42] A. Kiayias and M. Yung. Group signatures with efficient concurrent join. In R. Cramer, editor, *Advances in Cryptology — EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 198–214. Springer-Verlag, 2005. (Cited on page 26, 27.)

[43] A. Kiayias and H. Zhou. Concurrent blind signatures without random oracles. In *SCN 2006*, volume 4116 of *LNCS*, pages 49–62. Springer-Verlag, 2006. (Cited on page 26.)

[44] A. Kiayias and H. Zhou. Equivocal blind signatures and adaptive uc-security. In R. Canetti, editor, *Theory of Cryptography Conference, TCC 2008*, volume 4948 of *LNCS*, pages 340–355. Springer-Verlag, 2008. (Cited on page 3, 25, 26.)

[45] N. Koblitz and A. Menezes. Another look at generic group. IACR ePrint Archive 2006/230, 2006. (Cited on page 3.)

[46] T. Okamoto. Efficient blind and partially blind signatures without random oracles. In S. Halevi and T. Rabin, editors, *Theory of Cryptography Conference, TCC 2006*, volume 3876 of *LNCS*, pages 80–99. Springer-Verlag, 2006. Full version avaialble on ePrint archive. (Cited on page 25, 26.)

[47] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In J. Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, volume 1592 of *LNCS*, pages 223–238. Springer-Verlag, 1999. (Cited on page 25.)

[48] T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In J. Feigenbaum, editor, *Advances in Cryptology — CRYPTO '91*, volume 576 of *LNCS*, pages 129–140. Springer-Verlag, 1992. (Cited on page 24.)

[49] M. Rückert and D. Schröder. Security of verifiably encrypted signatures and a construction without random oracles. IACR ePrint Archive 2009/027, 2009. (Cited on page 2.)

[50] A. Rupp, G. Leander, E. Bangerter, A.-R. Sadeghi, and A. W. Dent. Sufficient conditions for intractability over black-box groups: Generic lower bounds for generalizaed dl and dh problems. IACR ePrint Archive 2007/360, 2007. (Cited on page 34.)

[51] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27(4), 1980. (Cited on page 33, 34.)

[52] M. Scott. Authenticated id-based key exchange and remote log-in with simple token and pin number. Cryptology ePrint Archive, Report 2002/164, 2002. `http://eprint.iacr.org/`. (Cited on page 4.)

[53] V. Shoup. Lower bounds for discrete logarithms and related problems. In W. Fumy, editor, *Advances in Cryptology — EUROCRYPT '97*, volume 1233 of *LNCS*, pages 256–266. Springer-Verlag, 1997. (Cited on page 3, 33, 34.)

[54] E. R. Verheul. Evidence that xtr is more secure than supersingular elliptic curve cryptosystems. *J. Cryptology*, 17(4):277–296, 2004. (Cited on page 4.)

Appendices

A Proofs Related to Assumptions

A.1 Proof of Theorem 1 ($\mathbf{DDH}_{\mathbb{G}_1} \Rightarrow \mathbf{DBP}$)

Proof. Assume that the DBP assumption does not hold and there is an adversary $\mathcal{A}$ that produces a pair $(z, r) \neq (1, 1)$ satisfying the equation $e(g_z, z)\ e(g_r, r) = 1$ for randomly chosen g_z, g_r with non-negligible probability. We construct $\mathcal{B}$ which breaks the DDH assumption in $\mathbb{G}_1$.

The $\mathrm{DDH}_{\mathbb{G}_1}$ assumption says that given a tuple $e(g, g_a, g_b, g_c) \in \mathbb{G}_1^4$, where $g_a = g^a$, $g_b = g^b$, and $g_c = g^c$, for $a, b, c \in \mathbb{Z}_p^*$ it is hard to distinguish distinguish between $c = ab$ and $c \neq ab$ with non-negligible probability. For a challenge tuple (g, g_a, g_b, g_c), $\mathcal{B}$ chooses a random $\psi \in \mathbb{Z}_p^*$ and gives to $\mathcal{A}$ an input (g^ψ, g_a^ψ) along with the appropriate public parameters. If $\mathcal{A}$ outputs $(z, r) \neq (1, 1)$ satisfying $e(g^\psi, z)\, e(g_a^\psi, r) = 1$, it is true that $z = r^{-a}$. Then, $e(g_b, z)\, e(g_c, r) = e(g^b, r^{-a})\, e(g^c, r) = e(g, r)^{c-ab}$; that equation is equal to 1 if and only $ab = c \mod p$.

Therefore, $\mathcal{B}$ has the same success probability of breaking the $\mathrm{DDH}_{\mathbb{G}_1}$ assumption as $\mathcal{A}$ of breaking the DBP assumption. ∎

A.2 Proof of Theorem 3 (SFP $\Rightarrow$ SDP)

Proof. Suppose that there exists an algorithm, $\mathcal{A}$, that successfully finds (z, r, u) that fulfills (2). We construct an algorithm that breaks SFP as follows. Given an SFP instance $(\Lambda, g_z, h_z, g_r, h_u, a, \tilde{a}, b, \tilde{b}, R_1, \ldots, R_q)$, input (g_z, h_z, g_r, h_u) to $\mathcal{A}$. If $\mathcal{A}$ outputs $(z^\star, r^\star, u^\star)$, set $(s^\star, t^\star, v^\star, w^\star) = (a, \tilde{a}, b, \tilde{b})$ and output $R^\star = (z^\star, r^\star, s^\star, t^\star, u^\star, v^\star, w^\star)$.

Now, multiplying $1 = e(g_z, z^\star), e(g_r, r^\star)$ to both sides of $(a, \tilde{a}) = e(s^\star, t^\star)$ results in the first equation in (3). Similarly, multiplying $1 = e(h_z, z^\star), e(h_u, u^\star)$ to both sides of $(b, \tilde{b}) = e(v^\star, w^\star)$ results in the second equation in (3). Thus $R^\star$ fulfills relations in (3). Since $(z^\star, r^\star, u^\star)$ is a valid answer to SDP, $z^\star \neq 1$ holds. Since every z_j in R_j is uniformly chosen and independent of $(g_z, h_z, g_r, h_u, a, \tilde{a}, b, \tilde{b})$, it is independent of the view of the adversary. Thus $z^\star = z_j$ happens only with negligible probability for every $j \in \{1, \ldots, q\}$. Thus $R^\star$ is a correct and valid answer to the SFP instance. ∎

A.3 Proof of Theorem 2 and Theorem 6 (Justification of SFP and k-SFP)

Proof.

Outline. In the generic model, every group element is represented by a unique index. The group operation to two group elements corresponds to addition of two indices. In the simulation, the index for an independent group element is represented by a unique variable. The index for a group element that is related to independent group elements is represented by a function of the variables determined by the relation.

A security argument in the generic group model consists of three steps. First we argue that no linear combinations of indices of initially given group elements could yield a new set of indices that fulfills the target predicate implied by the assumption. This step is done by inspecting the form of possible representation of indices. Although the argument looks intricate and lengthy, the underlying idea follows the standard approach.

The second step is to estimate the success probability of the adversary when uniform assignment is done to the variables. The adversary is considered as successful either when the simulation happens to be inconsistent to the concrete assignment or when the output of the adversary happens to fulfill the target predicate. For this, we estimate the probability that two indices represented by functions of variables are not identical but fall into the same value when concrete values are assigned to the variables. A common idea for this step is to apply Schwartz's lemma [53, 51] to the formula representing the difference between two indices. When the formula is a polynomial, it promptly gives an upper bound to the probability due to the degree of the polynomial. In the case of SDH, however, the formula will be in the form of $\frac{1}{x-c_1} + \cdots + \frac{1}{x-c_q}$ that results in having x^q in the polynomial expression of the formula. Thus it gives an upper bound with factor of q. By accumulating the error probability for all combinations of indices that could be generated through ℓ group operations, total error probability is bound by $\mathcal{O}(q\ell^2)$, which is apart from the optimal $\mathcal{O}(\ell^2)$ bound in DL and CDH [53] by the factor of q. In the case of SFP, the formula is in the form of $u_1 + \frac{1}{u_1} + \cdots + u_q + \frac{1}{u_q}$, which is a Laurent polynomial. Directly applying Schwartz's lemma as introduced in [53] or its variation for Laurent polynomial in [50] results in loose bound with factor of q since the formula is in degree q in its (regular) polynomial form. We instead follow the *proof idea* of the original Schwartz's lemma as introduced in [51] and show a constant bound with factor of 2. In the actual argument, we have to consider products of two indices to take care of pairing operations, which makes the analysis more intricate. But we can show a constant bound in the same way. As a result, the factor of q is eliminated and we have optimal $\mathcal{O}(\ell^2)$ bound.

Details. SFP is a special case of k-SFP at $k = 1$. For simplicity, we start with the case of SFP and then show how to generalize the argument to the case of k-SFP. We consider the generic group model for symmetric group setting where $\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G}$. The symmetric setting is not just for simplicity but also for generality as our argument in the symmetric setting trivially holds in the asymmetric setting as well. (The error probability can be slightly improved in the asymmetric setting since each group has less elements.)

In the following, we focus on the relation between the indexes of the group elements by translating the assumption appropriately. To make the argument easily linkable to its original representation, we use the same letter to denote the index of a group element with respect to an implicit fixed base. For instance, for $a \in \mathbb{G}$, we denote $\log a \in \mathbb{Z}_p$ by a itself. By using this notation, SFP is translated as follows. Let I be a tuple that consists of $(g_z, h_z, g_r, h_u, a, \tilde{a}, b, \tilde{b})$ and $R_j = (z_j, r_j, s_j, t_j, u_j, v_j, w_j)$ for $j = 1, \ldots, q$, which in total consists of $8 + 7q$ variables. Recall R_j fulfills relations

$$A \stackrel{\text{def}}{=} a \cdot \tilde{a} = g_z \cdot z_j + g_r \cdot r_j + s_j \cdot t_j, \quad \text{and} \tag{24}$$

$$B \stackrel{\text{def}}{=} b \cdot \tilde{b} = h_z \cdot z_j + h_u \cdot u_j + v_j \cdot w_j. \tag{25}$$

Let $O = (z^\star, r^\star, s^\star, t^\star, u^\star, v^\star, w^\star)$ be a 7-tuple of linear combinations of the variables in I. We show that there is no O that identically fulfills relations

$$A = g_z \cdot z^\star + g_r \cdot r^\star + s^\star \cdot t^\star, \quad \text{and} \tag{26}$$

$$B = h_z \cdot z^\star + h_u \cdot u^\star + v^\star \cdot w^\star \tag{27}$$

under the constraint that $z^\star \notin \{0, z_1, \ldots, z_q\}$.

For a polynomial $x \in \mathbb{Z}_p[I]$ of degree 1, let $\psi(x)$ be the linear form (line matrix of $1 \times |I|$) associated to x. Then, for $x, y \in \mathbb{Z}_p[I]$, product $x \cdot y$ is represented by a $|I| \times |I|$ symmetric matrix, say $\Psi(x \cdot y)$,

$$\Psi(x \cdot y) = \frac{1}{2} \left(\psi(x) \circ \psi(y)^T + \psi(y) \circ \psi(x)^T \right). \tag{28}$$

It is important to see that $\text{Rank}(\Psi(x \cdot y)) = 2$. Each row and column is associated to a variable, e.g., a in I and called a-row and a-column. By $\Psi(X + Y)$ we denote a matrix obtained by $\Psi(X) + \Psi(Y)$. In the following, we re-write $a, \tilde{a}, b, \tilde{b}$ by s_0, t_0, v_0, w_0, respectively, and let $z_0 = r_0 = u_0 = 0$ for seamless argument. Define A_j and B_j as $A_j = \Psi(g_z \cdot z_j + g_r \cdot r_j + s_j \cdot t_j)$ and $B_j = \Psi(h_z \cdot z_j + h_u \cdot r_j + v_j \cdot w_j)$ for $j = 0, \ldots, q$.

Relation (26) holds if and only if $\Psi(g_z \cdot z^\star + g_r \cdot r^\star + s^\star \cdot t^\star)$ is identical to a quadratic form representation of A. Since $\text{Rank}(\Psi(g_z \cdot z^\star + g_r \cdot r^\star + s^\star \cdot t^\star)) \leq 6$, it suffices to consider a quadratic form representation of A whose rank is less than 6. Recall that A is defined by A_0 and equivalently by A_j for $j = 1, \ldots, q$. For some symmetric matrix X of size $|I| \times |I|$, Consider $R = (((X \bmod A_0) \bmod A_1) \ldots)$. If $R \neq \emptyset$, there exists an assignment to the variables that evaluates the quadratic form associated by R to a non-zero value. Accordingly, for a matrix X to be evaluated to A for any valid assignments to the variables, X must be identical to $\sum c_j A_j$ where $\sum c_j = 1$ for some constant $c_j \in \mathbb{Z}_p$. Observe that $\mathrm{Rank}(A_j) = 6$ for $j = 1, \ldots, q$. Also observe that for any $j = 0, \ldots, q$, and for any $j' \neq j$, it holds that $\mathrm{Rank}(c_j A_j + c_{j'} A_{j'}) \geq 8$ when $c_i \neq 0$ and $c_{i'} \neq 0$. Thus only individual A_j for $j = 0, \ldots, q$ are the quadratic form representations of A with rank equal to or less than 6. It therefore suffices to consider A_j for the left hand of (26). By the same argument, it suffices to consider B_j for the left hand of (27).

Observe that $\mathrm{Rank}(A_j) = 6$ for $j = 1, \ldots, q$. Also observe that for any $j = 0, \ldots, q$, and for any $j' \neq j$, s_j-row and t_j-row in A_j are linearly independent from $A_{j'}$ and these rows are zeros in $A_{j'}$. Accordingly, $\mathrm{Rank}(c_j A_j + c_{j'} A_{j'}) \geq 8$ if $c_i \neq 0$ and $c_{i'} \neq 0$. Thus only individual A_j for $j = 0, \ldots, q$ are quadratic form representation of A with rank equal to or less than 6. It therefore suffices to consider A_j for the left hand of (26). The same argument applies to (27).

Suppose that (26) holds, namely:

$$\Psi(g_z \cdot z_j) + \Psi(g_r \cdot r_j) + \Psi(s_j \cdot t_j) = \Psi(g_z \cdot z^\star) + \Psi(g_r \cdot r^\star) + \Psi(s^\star \cdot t^\star) \tag{29}$$

holds. Observe that all cells in matrix $\Psi(g_z \cdot z^\star)$ are zeros except for those in the g_z-row and g_z-column. Thus $\Psi(s_j \cdot t_j)$ is not covered by $\Psi(g_z \cdot z^\star)$. Similarly, $\Psi(s_j \cdot t_j)$ is not covered by $\Psi(g_r \cdot r^\star)$, either. Thus $\Psi(s^\star \cdot t^\star)$ covers $\Psi(s_j \cdot t_j)$. Observe that $\mathrm{Rank}(\Psi(s^\star \cdot t^\star)) = 2$ and the s_j-row and t_j-row in $\Psi(s^\star \cdot t^\star)$ are linearly independent each other. Since $z^\star \neq z_j$, we have either $z^\star = c \cdot z_j$ for some constant $c \neq 1$ or $z^\star$ is a linear combination of variables including at least one variable other than z_j. In either case, the g_z-row of $\Psi(g_z \cdot z^\star)$ is linearly dependent on the s_j-row and t_j-row in $\Psi(s^\star \cdot t^\star)$. This results in having non-zero terms of s_j or t_j or both in $z^\star$. Without loss of generality, assume that $z^\star$ includes a term in s_j. Now consider the other relation

$$\Psi(h_z \cdot z_{j'}) + \Psi(h_u \cdot r_{j'}) + \Psi(v_{j'} \cdot w_{j'}) = \Psi(h_z \cdot z^\star) + \Psi(h_u \cdot r^\star) + \Psi(v^\star \cdot w^\star) \tag{30}$$

where j' can be different from j in (29). Since $z^\star$ includes s_j, cell (h_z, s_j) is non-zero in $\Psi(h_z \cdot z^\star)$. Since the cell is zero in the matrices in the left hand of (30), it must be offset by $\Psi(h_u \cdot r^\star) + \Psi(v^\star \cdot w^\star)$. In $\Psi(h_u \cdot r^\star)$, the cell is zero since only h_u-row and h_u-column can have non-zero cells. Therefore cell (h_z, s_j) is non-zero in $\Psi(v^\star \cdot w^\star)$. Due to the same reason as before, $\Psi(v^\star \cdot w^\star)$ must cover $\Psi(v_{j'} \cdot w_{j'})$ in (30). Since $\mathrm{Rank}(\Psi(v_{j'} \cdot w_{j'}))$ is two, the h_z-row must be linearly dependent on $v_{j'}$-row and $w_{j'}$-row. Thus either $(v_{j'}, s_j)$ or $(w_{j'}, s_j)$ must be non-zero. However, none of the matrices in (30) has non-zero value in $(v_{j'}, s_j)$ and $(w_{j'}, s_j)$. Thus (30) cannot hold. This completes the case of $k = 1$.

To generalize the above argument to the case of $k \geq 2$, simply replace $\Psi(s_j \cdot t_j)$ with $\Psi(\sum_{i=1}^{k} s_{ji} \cdot t_{ji})$ and argue in the same way based on the observation that $\mathrm{Rank}(\Psi(\sum_{i=1}^{k} s_{ji} \cdot t_{ji})) = 2k$.

We now proceed to evaluate the error probability of the generic group oracle simulation for the case of general k. Namely, we consider the probability, say P_1, that two distinct elements in $\mathbb{G}$ evaluates to the same value by assignment. An index for an element in $\mathbb{G}$ is a linear combination of variables in I. For index f and f' of two distinct elements in $\mathbb{G}_T$, probability P_1 is $\Pr[f - f' = 0]$. Here the probability is taken over uniform assignments to the independent variables in I. Among the variables in I we consider r_j and u_j be dependent and represent them by

$$r_j = \left(\sum_{i=1}^{k} (a_i \cdot \tilde{a}_i - s_{ji} \cdot t_{ji}) - g_z \cdot z_j \right) / g_r, \text{ and} \tag{31}$$

$$u_j = \left(\sum_{i=1}^{k} \left(b_i \cdot \tilde{b}_i - v_{ji} \cdot w_{ji} \right) - h_z \cdot z_j \right) / h_u. \tag{32}$$

Let F be a polynomial obtained by replacing r_j and u_j in f with the right hands of the above equations and multiplying $g_r \cdot h_u$. Define F' for f' in the same way. Then we have $P_1 = \Pr[F - F' = 0]$. Since $\deg(F - F') = 3$, we have $P_1 \leq 3/p$ from Schwartz's lemma. Having initial $|I| = 2 + 4k + (3 + 4k)q$ elements and at most ℓ_1 group operations, we have the upper bound

$$\binom{2 + 4k + (3 + 4k)q + \ell_1}{2} \cdot 3/p \leq O((k \cdot q + \ell_1)^2)/p \tag{33}$$

for the simulation error of elements in $\mathbb{G}$.

The error probability, say P_T, in simulating $\mathbb{G}_T$ is estimated in the similar way. An index of an element in $\mathbb{G}_T$ is in a quadratic form of variables in I. Let F_T and F'_T be polynomials obtained from the indexes of two distinct elements in $\mathbb{G}_T$ in the same way as above. Then $P_T = \Pr[F_T - F'_T = 0]$. Since $\deg(F_T - F'_T) = 4$, we have $P_T \leq 4/p$ from Schwartz's lemma. Having at most ℓ_T pairing operations and group operations in $\mathbb{G}_T$, at most ℓ_T elements appear in $\mathbb{G}_T$ during the simulation. Thus the error probability throughout the simulation is bound by $\binom{\ell_T}{2} \cdot P_T = \mathcal{O}(\ell_T^2)/p$.

In total, the simulation error is upper bound by $P_1 + P_T = \mathcal{O}(\ell_T^2 + (k \cdot q + \ell_1)^2)/p$. By setting $\ell = \ell_1 \approx \ell_T$, it is simplified to $\mathcal{O}(k^2 \cdot q^2 + \ell^2)/p$ as stated in Theorem 6. Setting $k = 1$ gives $\mathcal{O}(q^2 + \ell^2)/p$ as in Theorem 2. ∎

B Homomorphic Trapdoor Commitment Schemes

This section presents several homomorphic trapdoor commitment schemes in bilinear settings. Not all of them are used in our construction. Nevertheless, we introduce them because they have different properties and may be useful in applications needing specific properties. We note that all the schemes in this section can also work as a chameleon hash. Namely, it is possible to equivocate any commitment generated by TC.Com rather than the ones simulated by TC.Sim. Indeed, we integrate TC1 as a chameleon hash in the construction of SSIG in Section 6.

By $(\mathcal{K}, \mathcal{M}, \mathcal{C}, \mathcal{D})$ we denote spaces for commitment-keys, messages, commitments, and decommitments. The commitment-key refers to elements not included in Λ. Table 3 shows a summary of the schemes in their space parameters and performance in verifying the correct opening. For comparison, we list schemes from [37] and [26] which are the only homomorphic trapdoor commitment schemes we aware in the literature whose messages are group elements and the verification is done by checking pairing product equations.

TC3 and [26] are GS-compatible schemes whose components are all in the base groups. In particular, TC3 is the first multi-commitment scheme that commits to k elements at a time. Its commitment has $2k + 2$ group elements while it will be $3k$ if we repeatedly use [26] for k times. It is an interesting open problem to construct a constant-size commitment scheme while being compatible with GS-proofs.

Scheme TC4 is independently found in [38], the updated version of [37], and is included in [1].

Scheme	Λ	$\mathcal{K}$	$\mathcal{M}$	$\mathcal{C}$	$\mathcal{D}$	#(pairings)	#(PPE)	assumption
$\mathsf{TC1}$	any	$2k+2^{[1]}$	$k^{[2]}$	$2^{[T]}$	$2^{[2]}$	$2k+2$	2	SDP $\ll$ DLIN
$\mathsf{TC4}$	$\Lambda_{\mathsf{xdh}}, \Lambda_{\mathsf{sxdh}}$	$k+1^{[1]}$	$k^{[2]}$	$1^{[T]}$	$1^{[2]}$	$k+1$	1	DBP $\ll$ DDH$_{\mathbb{G}_1}$
Gro09[37]	any	$2k+4^{[1]}$	$k^{[2]}$	$2^{[T]}$	$2^{[2]}$	$2k+4$	2	STP $\ll$ DLIN
$\mathsf{TC3}$	Λ_{sym}	$2k+2^{[1]}$	$k^{[1]}$	$2k+2^{[1]}$	$2^{[1]}$	2k+ 2	2	SDP $\ll$ DLIN
CLY09[26]	Λ_{sym}	$5^{[1]}$	$1^{[1]}$	$3^{[1]}$	$3^{[1]}$	9	3	DLIN
$\mathsf{TC2}$	any	$2^{[2]}$	$1^{[p]}$	$1^{[2]}$	$1^{[1]}$	2	1	XDHI ($\ll$ DLIN)

Table 3: Summary of homomorphic trapdoor commitments. Columns from $\mathcal{K}$ to $\mathcal{D}$ count the number of elements and indicating the groups they belong to ([1], [2], $[T]$, and $[p]$ respectively for $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$, and $\mathbb{Z}_p$). #(pairings) and #(PPE) count the number of pairings and pairing product equations in the verification predicate. On top are the multi-message schemes committing to k group elements at once; in the middle are the schemes *not* using any group element in $\mathbb{G}_T$; and at the bottom is the efficient scheme when the message is in $\mathbb{Z}_p$ and the other components are in $\mathbb{G}_1$ and $\mathbb{G}_2$. $X \ll Y$: Assumption X is implied by Y (if $\Lambda = \Lambda_{\mathsf{sym}}$).

For multi-message commitment schemes, $\mathsf{TC1}, \mathsf{TC3}, \mathsf{TC4}$, let $\vec{m} = \{m_1, \ldots, m_k\} \in {\mathbb{G}_2}^k$ be a message. For single-message commitment scheme, $\mathsf{TC2}$, let m be an element of $\mathbb{Z}_p$. In the following description, we assume that Λ is given to all algorithms implicitly.

B.1 Scheme TC1

$\mathsf{TC1.Key}(1^\lambda)$: Choose random generators g_r, h_u from $\mathbb{G}_1^*$. For $i = 1, \ldots, k$, choose γ_i and δ_i from $\mathbb{Z}_p^*$ and compute $g_i = g_r^{\gamma_i}$ and $h_i = h_u^{\delta_i}$. Output commitment-key $ck = (g_r, h_u, g_1, h_1, \ldots, g_k, h_k)$ and trapdoor $tk = (\gamma_1, \delta_1 \ldots, \gamma_k, \delta_k)$.

$\mathsf{TC1.Com}(ck, \vec{m})$: Choose r and u randomly from $\mathbb{G}_2$, and compute

$$C_1 = e(g_r, r) \prod_{i=1}^{k} e(g_i, m_i) \quad \text{and} \quad C_2 = e(h_u, u) \prod_{i=1}^{k} e(h_i, m_i). \tag{34}$$

Output commitment $c = (C_1, C_2)$ and decommit-key $dk = (r, u)$.

$\mathsf{TC1.Vrf}(ck, c, \vec{m}, dk)$: Parse c into (C_1, C_2) and dk into (r, u). Output 1 if (34) holds. Output 0, otherwise.

$\mathsf{TC1.Sim}(ck)$: Choose r and u randomly from $\mathbb{G}_2$ and compute $C_1 = e(g_r, r)$ and $C_2 = e(h_u, u)$. Output commitment $c = (C_1, C_2)$ and equivocation-key $ek = (r, u)$.

$\mathsf{TC1.Equiv}(ck, \vec{m}, ek, tk)$: Parse ek into (r, u) and tk into $(\gamma_1, \delta_1 \ldots, \gamma_k, \delta_k)$. Then compute $r' = r \cdot \prod_{i=1}^{k} m_i^{-\gamma_i}$, and $u' = u \cdot \prod_{i=1}^{k} m_i^{-\delta_i}$. Then output decommit-key $dk = (r', u')$.

The above scheme shares many similarities with that of Groth [37], but the security is based on a different computational assumption, i.e., SDP. It should be noted that both assumptions are implied by DLIN.

Theorem 11. *Trapdoor commitment scheme TC1 is perfectly hiding and computationally binding under the SDP assumption.*

Proof. For perfect hiding, observe that, for any $(C_1, C_2) \in \mathbb{G}_T^2$, any $\vec{m} \in \mathbb{G}_2^k$, there exits a unique $(r, u) \in \mathbb{G}_2^2$ that fulfills relation (34).

For computational binding, suppose that there exists an adversary that successfully opens a commitment to two distinct messages. We show that one can break SDP by using such an adversary. Given an instance of SDP, $(\Lambda, g_r, h_u, g_z, h_z)$, do as follows.

- Set $g_i = g_z^{\chi_i} g_r^{\gamma_i}$ and $h_i = h_z^{\chi_i} h_u^{\delta_i}$ for $i = 1, \ldots, k$. Run the adversary with $ck = (g_r, h_u, \{g_i, h_i\}_{i=1}^k)$.
- Given two openings $(\vec{m}, r, u)$ and $(\vec{m}', r', u')$ from the adversary, compute

$$z^\star = \prod_{i=1}^k \left(\frac{m_i}{m'_i}\right)^{\chi_i}, \quad r^\star = \frac{r}{r'} \prod_{i=1}^k \left(\frac{m_i}{m'_i}\right)^{\gamma_i}, \quad u^\star = \frac{u}{u'} \prod_{i=1}^k \left(\frac{m_i}{m'_i}\right)^{\delta_i}. \tag{35}$$

- Output $(z^\star, r^\star, u^\star)$.

Since the openings fulfills (34), we have

$$\begin{aligned} 1 &= e\left(g_r, \frac{r}{r'}\right) \prod e\left(g_i, \frac{m_i}{m'_i}\right) = e\left(g_z, \prod_{i=1}^k \left(\frac{m_i}{m'_i}\right)^{\chi_i}\right) e\left(g_r, \frac{r}{r'} \prod_{i=1}^k \left(\frac{m_i}{m'_i}\right)^{\gamma_i}\right) \\ &= e(g_z, z^\star)\, e(g_r, r^\star), \text{ and} \\ 1 &= e\left(h_u, \frac{u}{u'}\right) \prod e\left(h_i, \frac{m_i}{m'_i}\right) = e\left(h_z, \prod_{i=1}^k \left(\frac{m_i}{m'_i}\right)^{\chi_i}\right) e\left(h_u, \frac{u}{u'} \prod_{i=1}^k \left(\frac{m_i}{m'_i}\right)^{\delta_i}\right) \\ &= e(h_z, z^\star)\, e(h_u, u^\star). \end{aligned}$$

But $\vec{m} \neq \vec{m}'$, so there exists i such that $m_i/m'_i \neq 1$. Also, χ_i is independent from the view of the adversary. That is, for every choice of χ_i, there exist corresponding γ_i and δ_i that gives the same g_i and h_i. Therefore, $z^\star = \prod_i (m_i/m'_i)^{\chi_i} \neq 1$ with overwhelming probability. Hence $(z^\star, r^\star, u^\star)$ is a valid answer to the instance of SDP. ∎

B.2 Scheme TC2

Let $g \in \mathbb{G}_1$ and $\tilde{g} \in \mathbb{G}_2$ be random bases. Common parameter Λ is given to all algorithms described below.

- $\mathsf{TC2.Key}(1^\lambda)$: Select $\gamma \in \mathbb{Z}_p$ and set $\tilde{f} = \tilde{g}^\gamma$. Output commitment key $ck = (\Lambda, \tilde{f})$ and trapdoor $tk = \gamma$.
- $\mathsf{TC2.Com}(ck, m)$: Choose random $\delta \in \mathbb{Z}_p$ and compute commitment $c = \tilde{g}^m \tilde{f}^\delta \in \mathbb{G}_2$ and decommit-key $d = g^\delta \in \mathbb{G}_1$. Output c and d.
- $\mathsf{TC2.Vrf}(ck, c, m, d)$: Output 1 if $e(g, c/\tilde{g}^m) = e(d, \tilde{f})$. Output 0, otherwise.
- $\mathsf{TC2.Sim}(ck)$: Choose random $\delta \in \mathbb{Z}_p$ and output a commitment $c = \tilde{f}^\delta$ and an equivocation-key $ek = \delta$.
- $\mathsf{TC2.EqOpen}(ck, m, ek, tk)$: Let $\delta = ek$ and $\gamma = tk$. Output $d = g^{\delta - m/\gamma}$.

The correctness follows since

$$e(g, c/\tilde{g}^m) = e(g, \tilde{f}^\delta) = e(g^\delta, \tilde{f}) = e(d, \tilde{f}).$$

The trapdoor property holds because

$$e(d, \tilde{f}) = e(g^{\delta - m/\gamma}, \tilde{f}) = e(g, \tilde{g}^{-m}\tilde{f}^\delta) = e(g, c/\tilde{g}^m).$$

To prove computational binding property, we assume that the following variant of Diffie-Hellman inversion problem (XDHI) is hard with respect to Λ.

Assumption 5 (XDHI). Given Λ and $(g, \tilde{g}, \tilde{g}^a) \in \mathbb{G}_1^* \times \mathbb{G}_2^* \times \mathbb{G}_2^*$ for random $a \in \mathbb{Z}_p^*$, it is hard to compute $g^{1/a} \in \mathbb{G}_1$.

Depending on setting Λ, the XDHI assumption is implied by basic assumptions, Computational Diffie-Hellman Assumption (CDH), Computational Co-Diffie-Hellman Assumption (co-CDH) [15], and Decisional Diffie-Hellman Assumption in $\mathbb{G}_2$ ($\mathrm{DDH}_{\mathbb{G}_2}$), as follows. Note that, CDH is implied by DLIN in Λ_{sym} and $\mathrm{DDH}_{\mathbb{G}_2}$ is implied by SXDH in Λ_{sxdh}.

Lemma 4. *CDH $\Rightarrow$ XDHI for Λ_{sym}. co-CDH $\Rightarrow$ XDHI for Λ_{xdh}. $\mathrm{DDH}_{\mathbb{G}_2}$ $\Rightarrow$ XDHI for Λ_{sxdh}.*

Proof. Let $\mathcal{A}$ be an XDHI adversary. In Λ_{sym}, given an CDH instance (g, g^α, g^β), input (g^α, g^β, g) to $\mathcal{A}$. It outputs $g^{\alpha\beta}$, which is the answer to the CDH instance. Next, in Λ_{xdh}, given an co-CDH instance $(g, g^\alpha, \tilde{g}, \tilde{g}^\beta) \in \mathbb{G}_2^{*3}$, input $(g^\alpha, \tilde{g}^\beta, \tilde{g})$ to $\mathcal{A}$. It outputs $g^{\alpha\beta}$, which is the answer to the co-CDH instance. In Λ_{sxdh}, observe that, on input $(g, \tilde{g}^x, \tilde{g})$, adversary $\mathcal{A}$ outputs g^x. Thus $\mathcal{A}$ provides a mapping from $\mathbb{G}_2$ to $\mathbb{G}_1$. Now, given an instance $(\tilde{g}, \tilde{g}^\alpha, \tilde{g}^\beta, \tilde{g}^\gamma)$ of $\mathrm{DDH}_{\mathbb{G}_2}$, input $(g, \tilde{g}^\alpha, \tilde{g})$ to $\mathcal{A}$ and receive g^α. Then $\gamma = \alpha\beta$ can be tested by checking if $e(g^\alpha, \tilde{g}^\beta) = e(g, \tilde{g}^\gamma)$ holds or not. ∎

Theorem 12. *Trapdoor commitment scheme $\mathsf{TC2}$ is perfectly hiding. It is binding if the XDHI assumption holds for Λ.*

Proof. The perfect hiding property holds from the fact that, for any $c \in \mathbb{G}_2$, for every $m \in \mathbb{Z}_p$ there exists a single consistent $\delta \in \mathbb{Z}_p$.

The binding property is proven by showing a reduction to XDHI. Given an instance of XDHI, $(g, \tilde{g}, \tilde{g}^a)$, set $\tilde{f} = \tilde{g}^a$. Suppose that an adversary outputs a commitment c correctly opened to (m, d) and (m', d') for $m \neq m'$. Then $e(g, c/\tilde{g}^m) = e(d, \tilde{f})$ and $e(g, c/\tilde{g}^{m'}) = e(d', \tilde{f})$ hold. By dividing both sides of the equations, we have $e(g, \tilde{g}^{m-m'}) = e(d'/d, \tilde{f}) = e(d'/d, \tilde{g}^a)$. Thus $(d'/d)^{1/m-m'} = g^{1/a}$, which is a correct answer to the XDHI instance. ∎

B.3 Scheme $\mathsf{TC3}$

All components for this scheme is in $\mathbb{G}_1$ and $\mathbb{G}_2$. The underlying idea is to use $\mathsf{TC1}$ and, instead of publishing a commitment in $\mathbb{G}_T$, we publish the decommit-key and the message in a randomized way by applying the one-side randomization $\mathsf{RandOneSide}$ from Section 3.

$\mathsf{TC3.Key}(1^\lambda)$: Choose random generators g_r, h_u from $\mathbb{G}_2^*$. For $i = 1, \ldots, k$, choose γ_i and δ_i from $\mathbb{Z}_p^*$ and compute $g_i = g_r^{\gamma_i}$ and $h_i = h_u^{\delta_i}$. Output commitment-key $ck = (\Lambda, g_r, h_u, \ldots, g_k, h_k)$ and trapdoor $tk = (\gamma_1, \delta_1 \ldots, \gamma_k, \delta_k)$.

$\mathsf{TC3.Com}(ck, \vec{m})$: Choose r and u randomly from $\mathbb{G}_2$, and compute

$$\{c_{ai}\}_{i=0}^{k} \leftarrow \mathsf{RandOneSide}((g_r, r), (g_1, m_1), \ldots, (g_k, m_k)), \text{ and} \tag{36}$$

$$\{c_{bi}\}_{i=0}^{k} \leftarrow \mathsf{RandOneSide}((h_u, u), (h_1, m_1), \ldots, (h_k, m_k)). \tag{37}$$

Output commitment $c = (\{c_{ai}\}_{i=0}^{k}, \{c_{bi}\}_{i=0}^{k})$ and decommit-key $dk = (r, u)$.

$\mathsf{TC3.Vrf}(ck, c, \vec{m}, dk)$: Parse c into $(\{c_{ai}\}_{i=0}^{k}, \{c_{bi}\}_{i=0}^{k}) \in \mathbb{G}_2^{2k+2}$ and dk into $(r, u) \in \mathbb{G}_2^2$. Output 1 if they satisfy the following predicates. Output 0, otherwise.

$$1 = e(g_r, r/c_{a0}) \prod_{i=1}^{k} e(g_i, m_i/c_{ai}) \quad \text{and} \quad 1 = e(h_u, u/c_{b0}) \prod_{i=1}^{k} e(h_i, m_i/c_{bi}) \tag{38}$$

$\mathsf{TC3.Sim}(ck)$: Do the same as $\mathsf{TC3.Com}$ with $\vec{m} = (1, \ldots, 1)$ and set $ek = (r, u)$.

$\mathsf{TC3.Equiv}(ck, \vec{m}, ek, tk)$: Parse ek into (r, u) and tk into $(\gamma_1, \delta_1 \ldots, \gamma_k, \delta_k)$. Then compute $r' = r \cdot \prod_{i=1}^{k} m_i^{-\gamma_i}$, and $u' = u \cdot \prod_{i=1}^{k} m_i^{-\delta_i}$. Then output decommit-key $dk = (r', u')$.

Theorem 13. *Trapdoor commitment scheme $\mathsf{TC3}$ is perfectly hiding and computationally binding under the SDP assumption.*

The hiding property is clear from the uniform output property of $\mathsf{RandOneSide}$ and that of $\mathsf{TC1}$. The binding property is taken over from $\mathsf{TC1}$ and can be proven in the same way as for $\mathsf{TC1}$.

B.4 Scheme TC4

This is the most efficient scheme both in computation and storage. The scheme virtually 'half' the scheme of $\mathsf{TC1}$. Let $\Lambda \in \{\Lambda_{\mathsf{xdh}}, \Lambda_{\mathsf{sxdh}}\}$.

$\mathsf{TC4.Key}(1^\lambda)$: Choose random generators g_r from $\mathbb{G}_1^*$. For $i = 1, \ldots, k$, choose γ_i from $\mathbb{Z}_p^*$ and compute $g_i = g_r^{\gamma_i}$. Output commitment-key $ck = (\Lambda, g_r, g_1, \ldots, g_k)$ and trapdoor $tk = (\gamma_1, \ldots, \gamma_k)$.

$\mathsf{TC4.Com}(ck, \vec{m})$: Choose r randomly from $\mathbb{G}_2$, and compute

$$c = e(g_r, r) \prod_{i=1}^{k} e(g_i, m_i). \tag{39}$$

Output commitment c and decommit-key $dk = r$.

$\mathsf{TC4.Vrf}(ck, c, \vec{m}, dk)$: Output 1 if (39) holds. Output 0, otherwise.

$\mathsf{TC4.Sim}(ck)$: Choose r randomly from $\mathbb{G}_2$ and compute $c = e(g_r, r)$. Output commitment c and equivocation-key $ek = r$.

$\mathsf{TC4.Equiv}(ck, \vec{m}, ek, tk)$: Take r and $(\gamma_1, \ldots, \gamma_k)$ out from ek and tk, respectively. Then compute $r' = r \cdot \prod_{i=1}^{k} m_i^{-\gamma_i}$, and Then output decommit-key $dk = r'$.

Theorem 14. *$\mathsf{TC4}$ is perfectly hiding and computationally binding if the DBP assumption holds for Λ.*

Proof. The hiding property holds because, for any commitment $c \in \mathbb{G}_T$ and any $\vec{m} \in \mathbb{G}_2^k$, there exists consistent $t \in \mathbb{G}_2$ that fulfills relation (39).

The binding property is shown similarly to Theorem 11:
Given an instance of DBP, (Λ, g_z, g_r), do as follows.

- Set $g_i = g_z^{\chi_i} g_r^{\gamma_i}$. Run the adversary with $ck = (g_r, \{g_i\}_{i=1}^k)$.
- Given two openings $(\vec{m}, r)$ and $(\vec{m}', r')$ from the adversary, compute

$$z^\star = \prod_{i=1}^{k} (m_i/m_i')^{\chi_i}, \qquad r^\star = (r/r') \prod_{i=1}^{k} (m_i/m_i')^{\gamma_i}. \tag{40}$$

- Output $(z^\star, r^\star)$.

Since the openings fulfills (34), we have

$$\begin{aligned} 1 &= e\left(g_r, \frac{r}{r'}\right) \prod e\left(g_i, \frac{m_i}{m_i'}\right) = e\left(g_z, \prod_{i=1}^{k} \left(\frac{m_i}{m_i'}\right)^{\chi_i}\right) e\left(g_r, \frac{r}{r'} \prod_{i=1}^{k} \left(\frac{m_i}{m_i'}\right)^{\gamma_i}\right) \\ &= e(g_z, z^\star)\, e(g_r, r^\star). \end{aligned}$$

But $\vec{m} \neq \vec{m}'$, so there exists i such that $m_i/m_i' \neq 1$. Also, χ_i is independent from the view of the adversary. That is, for every choice of χ_i, there exist corresponding γ_i that gives the same g_i. Therefore, $z^\star = \prod_i (m_i/m_i')^{\chi_i} \neq 1$ with overwhelming probability. Hence $(z^\star, r^\star)$ is a valid answer to the instance of SDP. ∎

One can have a variant of TC4 whose commitment is in $\mathbb{G}_1$ and $\mathbb{G}_2$ in a similar way we convert TC1 to TC3. Unlike the previous case, however, RandOneSide cannot be used as TC4 is in $\Lambda = \Lambda_{\mathsf{sxdh}}$. So we instead use RandSeq keeping g_r and h_u intact. This modification results in $2k+1$ group elements in a commitment, which is 1 element less than that of TC3. However, depending on the applications, this may be less efficient since the verification predicate is not one-sided.

C One-Time Signature Schemes

C.1 A One-Time Signature Scheme in Any Setting

Let $\Lambda \in \{\Lambda_{\mathsf{sym}}, \Lambda_{\mathsf{xdh}}, \Lambda_{\mathsf{sxdh}}\}$.

- $\mathsf{OTS1.Key}(1^\lambda)$: Choose random generators $g_z, h_z, g_r, h_u \leftarrow \mathbb{G}_1^*$. For $i = 1, \ldots, k$, choose χ_i, γ_i, $\delta_i \leftarrow \mathbb{Z}_p$ and compute $(g_i, h_i) = (g_z^{\chi_i} g_r^{\gamma_i}, h_z^{\chi_i} h_u^{\delta_i})$. Also choose $\zeta, \rho, \varphi \leftarrow \mathbb{Z}_p$ and set $a = g_z^{\zeta} g_r^{\rho}$ and $b = h_z^{\zeta} h_u^{\varphi}$. Set $vk = (g_z, h_z, g_r, h_u, \{g_i, h_i\}_{i=1}^k, a, b)$ and $sk = (vk, \zeta, \rho, \varphi, \{\chi_i, \gamma_i, \delta_i\}_{i=1}^k)$. We also use $\tilde{g}$ in Λ. Output (vk, sk).
- $\mathsf{OTS1.Sign}(sk, \vec{m})$: Compute

$$z = \tilde{g}^{\zeta} \prod_{i=1}^{k} m_i^{-\chi_i}, \quad r = \tilde{g}^{\rho} \prod_{i=1}^{k} m_i^{-\gamma_i}, \quad u = \tilde{g}^{\varphi} \prod_{i=1}^{k} m_i^{-\delta_i}.$$

  Output $\sigma = (z, r, u)$ as a signature.

- $\mathsf{OTS1.Vrf}(vk, \vec{m}, \sigma)$: Parse σ into (z, r, u). Output 1 if the following equations hold. Output 0, otherwise.

$$e(a, \tilde{g}) = e(g_z, z)\, e(g_r, r) \prod_{i=1}^{k} e(g_i, m_i) \tag{41}$$

$$e(b, \tilde{g}) = e(h_z, z)\, e(h_u, u) \prod_{i=1}^{k} e(h_i, m_i) \tag{42}$$

Theorem 15. *One-time signature scheme OTS1 is strongly unforgeable against one-time chosen message attacks if SDP holds for Λ.*

Proof. Suppose that there is an adversary, $\mathcal{A}$, that finds a forged signature $\sigma^\dagger = (z^\dagger, r^\dagger, u^\dagger)$ for message $\vec{m}^\dagger$ after seeing a one-time signature (z, r, u) for message $\vec{m}$ of its choice. We construct a reduction algorithm to SDP as follows.

Given an instance (g_z, h_z, g_r, h_u) of SDP, do the same as $\mathsf{OTS1.Key}$ by using the input instance as the bases. When $\mathcal{A}$ submit message $\vec{m}$, run $\mathsf{OTS1.Sign}$ and return (z, r, u) to $\mathcal{A}$. Given output $(z^\dagger, r^\dagger, u^\dagger)$ and $\vec{m}^\dagger$ from $\mathcal{A}$, compute

$$z^\star = (z^\dagger/z) \prod_{i=1}^{k} (m_i^\dagger/m_i)^{\chi_i}, \quad r^\star = (r^\dagger/r) \prod_{i=1}^{k} (m_i^\dagger/m_i)^{\gamma_i}, \quad u^\star = (u^\dagger/u) \prod_{i=1}^{k} (m_i^\dagger/m_i)^{\delta_i}. \tag{43}$$

Then output $(z^\star, r^\star, u^\star)$. This completes the description of the reduction algorithm.

Suppose that adversary $\mathcal{A}$ is successful. By dividing both sides of (41) with respect to $(z^\star, r^\star, u^\star)$ and (z, r, u), we have

$$\begin{aligned} 1 &= e(g_z, z^\dagger/z)\, e(g_r, r^\dagger/r) \prod_{i=1}^{k} e(g_i, m_i^\dagger/m_i) \\ &= e(g_z, z^\dagger/z \prod_{i=1}^{k} (m_i^\dagger/m_i)^{\chi_i})\, e(g_r, r^\dagger/r \prod_{i=1}^{k} (m_i^\dagger/m_i)^{\gamma_i}) \\ &= e(g_z, z^\star)\, e(g_r, r^\star). \end{aligned}$$

Similarly, with respect to (42), we have

$$\begin{aligned} 1 &= e(h_z, z^\dagger/z)\, e(h_u, u^\dagger/u) \prod_{i=1}^{k} e(h_i, m_i^\dagger/m_i) \\ &= e(h_z, z^\dagger/z \prod_{i=1}^{k} (m_i^\dagger/m_i)^{\chi_i})\, e(h_u, u^\dagger/u \prod_{i=1}^{k} (m_i^\dagger/m_i)^{\delta_i}) \\ &= e(h_z, z^\star)\, e(h_u, u^\star). \end{aligned}$$

Hence $(z^\star, r^\star, u^\star)$ is a correct answer to the SDP instance.

What remains to show is $z^\star \neq 1$. We first consider the case of $\vec{m} = \vec{m}^\dagger$. In this case, $(z^\dagger, r^\dagger, u^\dagger) \neq (z, r, u)$ must hold. Observe that $z^\dagger = z$ cannot be the case since it implies $r^\dagger = r$ and $u^\dagger = u$ to fulfill (41) and (42). Thus we have $z^\dagger \neq z$ and $z^\star = z^\dagger/z \neq 1$. Next we consider the case of $\vec{m} \neq \vec{m}^\dagger$. In this case, there exists $i^\star$ for which $m_{i^\star} \neq m_{i^\star}^\dagger$ holds. For such $i^\star$, parameter

$\chi_{i^\star}$ is information theoretically hidden from the view of the adversary. Namely, for any view of the adversary and for any $\chi_{i^\star}$, there exists a consistent coin toss which yields the same view. This can be verified by seeing that (a, b), and (g_i, h_i) are perfectly hiding commitments of ζ and $\chi_{i^\star}$, and the one-time signature does not identify them despite establishing relation between them. Therefore, due to the term $(m^\dagger_{i^\star}/m_{i^\star})^{\chi_{i^\star}}$, for $m^\dagger_{i^\star} \neq m_{i^\star}$, the probability that $z^\star = 1$ is negligible. ∎

C.2 More Efficient Scheme in the Asymmetric Setting

In the case of $\Lambda \in \{\Lambda_{\mathsf{xdh}}, \Lambda_{\mathsf{sxdh}}\}$ we can construct a more efficient scheme, say $\mathsf{OTS2}$, that halves $\mathsf{OTS1}$ just like $\mathsf{TC4}$ does for $\mathsf{TC1}$. The verification equation is:

$$e(a, \tilde{g}) = e(g_z, z)\; e(g_r, r)\; \prod_{i=1}^{k} e(g_i, m_i) \tag{44}$$

Scheme $\mathsf{OTS2}$ is strongly unforgeable against one-time chosen message attacks under the DBP assumption.

C.3 Signing An Unbounded-Size Message

Using $\mathsf{OTS1}$ from Section C.1 we construct $\mathsf{OTS1u}$ that can sign unbounded-size message. (Thus it is an automorphic one-time signature scheme.) The idea is to sign a block of message together with a fresh verification-key used to sign the next message block. A problem is that the verification-key of $\mathsf{OTS1}$ is too large and not covered by its message space. We can get around the problem by reusing the bases $(g_z, h_z, g_r, h_u, \{g_i, h_i\}_{i=1}^{k})$ and only renew (a, b) for every message block. The same trick is used in Section 8. The unforgeability against one-time chosen message attacks can be proven based on SDP. The proof is almost the same as that for $\mathsf{OTS1}$ and omitted. (Since fresh a and b brings new randomness ζ, the information theoretic nature exploited in the proof is preserved.)

In the asymmetric case $\Lambda = \Lambda_{\mathsf{sxdh}}$, one can do the similar construction based on $\mathsf{OTS2}$. Since a is not in the message space, we use dual signature scheme as in Section 7 and sign messages in $\mathbb{G}_2$ and $\mathbb{G}_1$ in alternating manner.