Computer Standards & Interfaces 91 (2025) 103886

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Improving efficiency and security of Camenisch–Lysyanskaya signatures for anonymous credential systems

Jia-Chng Loh [*], Fuchun Guo, Willy Susilo

Institute of Cybersecurity and Cryptology, SCIT, University of Wollongong, Wollongong, Australia

ARTICLE INFO

Keywords:
Tight security reduction
Digital signatures
Randomizability
Anonymous credentials

ABSTRACT

Camenisch–Lysyanskaya signature scheme with randomizability, namely CL signatures, at CRYPTO'04 has been well adopted for many privacy-preserving constructions, especially in the context of anonymous credential systems. Unfortunately, CL signatures suffer from linear size drawbacks. The signature size grows linearly based on the signing messages, which decreases the interest in practice, as each user may have multiple attributes (messages). Its standard EUF-CMA security was first proven under an interactive assumption. While the interactive assumption is not desirable in cryptography, Fuchsbauer et al. revisited its security at CRYPTO'18 by proving the scheme under the discrete logarithm (Dlog) assumption in the algebraic group model (AGM) that idealizes the adversary's computation to be algebraic, yet the reduction loss is non-tight. In this work, we propose a new variant of CL signatures, namely CL+ signatures, that improves efficiency and security. The proposed CL+ signatures possess randomizability without the linear size drawback, such that signature size is a constant of three group elements. Besides, we prove the security of CL+ signatures can be tightly reduced to the DLog problem in AGM with only a loss factor of 3. Lastly, we show how CL+ signatures can also be instantiated to anonymous credential systems.

1. Introduction

Digital signature is a useful cryptographic primitive for many authentication systems. The randomizability is a very useful property for a signature scheme in a bilinear setting. Given a randomizable signature, one can simply generate a fresh signature with some random scalars on the same message. This randomizability due to Camenisch and Lysyanskaya, namely CL signature [1], has been adopted in many privacy-preserving constructions [2–8], especially the anonymous credential system [9], which enables users to privately obtain credentials and prove the possession of credentials without revealing users' information.

Although the CL signature scheme is known as a very flexible building block for many privacy-preserving constructions, one major drawback that limits being practical is that the signature size grows linearly based on the number of messages to be signed. This linearly signature size growth happens to be a practical issue in this case. For example, in the context of the anonymous credential system [10–13], users may have multiple attributes (n-messages) to be certified (signed). In particular, let n be the number of messages, the size of CL signatures will be $1 + 2n$ group elements.

The standard EUF-CMA security [14] of CL signatures was first proven under the LRSW assumption [1], which is an interactive assumption in the generic group model (GGM) [15,16]. In GGM, the adversary is oblivious to the group elements, and the group operations are mediated by the simulator except for checking the equality. Although one can design efficient schemes under interactive assumptions, these are not preferable [17,18]. In cryptography, it is always desirable to have schemes that can be provably secure under non-interactive, well-known assumptions, i.e. the discrete logarithm (DLog) problem, which is known as the hardest problem in the cyclic group setting.

Recently, Fuchsbauer et al. [19] introduced the algebraic group model (AGM) that idealizes the adversary's computation to be algebraic, such that any returned group element from the algebraic adversary must be described along with the representation based on all received group elements. Additionally, they showed for the first time how to prove the security of CL signatures under the DLog assumption in the standard EUF-CMA security model. However, the security reduction is non-tight even though the proof is in the AGM. A tight reduction is desired as it allows parameter size of a scheme in practice to be optimal — as efficient as proposed. Therefore, finding the possibility of having tight security for CL signatures under DLog assumption and EUF-CMA is still a challenging problem.

[*] Corresponding author.
E-mail address: jial@uow.edu.au (J.-C. Loh).

https://doi.org/10.1016/j.csi.2024.103886
Received 8 November 2023; Received in revised form 6 May 2024; Accepted 5 July 2024
Available online 8 July 2024

1.1. Our contribution

In this work, we propose a new variant of CL signature scheme, namely CL+ signature scheme, that resolves two key challenges. First, the proposed CL+ signatures possess identical features to CL signatures, but ours does not suffer the linear size drawback. Second, we propose a tight reduction algorithm that reduces the EUF-CMA security of CL+ signatures to the DLog problem in the algebraic group model (AGM). Furthermore, we also instantiate how we can build an anonymous credential system using the proposed CL+ signature scheme as the building block.

The proposed CL+ signature scheme is the first variant of CL signatures that can be tightly reduced to the standard DLog problem. In order to simplify the proof, we first transform the CL+ signature scheme into a single-message signature scheme, namely simpleCL+ signature scheme, and show a tight reduction is achieved through three simulations in the AGM, which have a success probability of $\frac{1}{3}$. This is in contrast to the original CL signatures in the AGM, in which their reduction can only succeed with a probability of $\frac{1}{3q}$, where q is the number of queries made. Later, we show a reduction that can reduce CL+ signature scheme to simpleCL+ signature scheme.

Lastly, we demonstrate that our proposed CL+ signature scheme can be employed to construct an anonymous credential system following CL's framework. This can be achieved by instantiating two protocols: (i) the protocol for signing committed messages and (ii) the protocol for proving knowledge of signatures. The former protocol enables one to obtain signatures without revealing the messages. While the latter protocol allows one to verify the signatures without revealing the plain message and signature pairs. In particular, these two protocols are built based on the (Generalized) Pedersen's commitment scheme and sigma protocol.

1.2. Related work

Randomizable Signatures. Pointcheval and Sanders at CT-RSA'16 [20] proposed an alternative CL-like scheme in the asymmetric-pairing setting, namely PS signatures, that preserves the same feature as in CL signatures, i.e., the randomizability and the flexibility to obtain an anonymous credential system. Additionally, PS signatures do not have the linear size drawback with short signature size: a constant of two group elements as compared to CL signatures that grow depending on the number of messages to be signed. However, its security was proven under the "modified" LRSW assumption [21] which holds under the asymmetric-pairing setting, a variant of LRSW assumption [22].

Since provable schemes under standard assumption are more desirable, Pointcheval and Sanders at CT-RSA'18 [23] revisited the security of the CL signatures [1] and PS signature [20]. They showed that instead of relying on the interactive assumption, both of the schemes can actually be provably secure under some variant of the q-SDH assumptions [24], namely q-Modified(M)SDH. However, these schemes can only be proven under EUF-(weak)CMA, which is a weaker security model such that the adversary can only query for signatures whose messages were pre-selected before the setup phase.

In order to achieve schemes under standard EUF-CMA, Pointcheval and Sanders [23] also presented modified CL signatures and PS signatures, respectively. These schemes release an additional element in $\mathbb{Z}_p$ as an extra signature element, which surprisingly improves the security to be EUF-CMA secure, but it weakens the randomizability as the extra signature element is not randomziable, which may not be desired. In order to achieve full-randomizability, one may derive such an element via a hash function, however, it results in the security of the scheme in the random oracle model, which idealizes the hash functions with truly random functions mediated by the simulator [25]. It is known that a provably secure scheme *without* random oracles is much preferable as random oracles cannot be implemented in practice [26,27].

We note that it is still unknown whether the security of the aforementioned schemes can be proven tightly reduced to the DLog problem in the AGM, though we realize PS signatures may be tightly reduced to symmetric DLog problem – a variant of DLog problem in asymmetric-pairing setting [7]. However, our main goal is to achieve a tightly secure scheme under the DLog assumption in the AGM.

Other Signature Schemes in AGM. Since the AGM was introduced, variants of signature schemes have been proven in this model. For instance, multi-signature schemes [28–32] enable multiple signers to collaborate in signing a single message; blind signature schemes [33–35] allow the signer to generate a legitimate signature without revealing the signing message; and threshold signature schemes [31,36] necessitate at least *t-out-of-n* signers to generate a valid signature together. These schemes achieve better efficiency when proven in AGM, either due to tightened reduction loss or a stronger security model.

Along the works for tight reduction under DLog assumption and standard EUF-CMA in the AGM, to the best of our knowledge, there are only Schnorr signatures [37], BLS signatures [38] and its identity-based setting, which are respectively proven in [19,33], and [39]. However, these schemes are not randomizable and their security is proven in AGM with random oracles [25].

The development of the AGM. The algebraic group model (AGM), formalized in [19], captures various computational hardness assumptions in the cyclic group settings, including the discrete logarithm (DLog) assumption and the computational Diffie–Hellman (CDH) assumption, along with its variants. Subsequent developments extended its scope to cover computational problems within the bilinear groups [40,41] and further refined it to handle decisional problems such as the decisional Diffie–Hellman (DDH) assumption [42]. Moreover, a framework was introduced in [41] aimed at capturing common hardness assumptions over (bilinear) groups. Therefore, providing opportunities to refine the security for some encryption schemes, e.g. the recent attribute-based encryption schemes [43,44].

On the other hand, the *strong* AGM was proposed [45] which provides a measurement of the number of group operations that an algebraic adversary has performed. This provides formal evidence backing the difficulty of the most commonly utilized time-lock puzzle [46]. Notably, recent findings demonstrate that the AGM can be instantiated under strong yet falsifiable assumptions within the standard model. This bolstering the AGM is a plausible model [47]

2. Preliminaries

Suppose $\mathbb{G}, \mathbb{G}_T$ are groups of prime order p. Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear pairing function, such that for all generator $g \in \mathbb{G}$, where $g \neq 1$, we have $e(g, g) \neq 1$ and $e(g^a, g^b) = e(g, g)^{ab}$. We say $param = (p, g, e, \mathbb{G}, \mathbb{G}_T)$ is a bilinear group tuple (system parameters) throughout the paper.

Definition 1 (*Bilinear Discrete Logarithm Assumption*)**.** The discrete logarithm (DLog) problem is hard in the bilinear pairing setting if given any random group element $g^a \in \mathbb{G}$, there is no probabilistic polynomial-time (PPT) algorithm can solve for $a \in \mathbb{Z}_p^*$ in polynomial time t with non-negligible advantage ϵ.

Definition 2 (*LRSW Assumption [22]*)**.** An interactive assumption that let $(g, X = g^x, Y = g^y)$ such that $x, y \in \mathbb{Z}_p^*$, and an oracle $\mathcal{O}(m)$ that on input $m \in \mathbb{Z}_p$ it returns a triple $T = (h, h^y, h^{x+m\cdot x\cdot y})$ such that $h \in \mathbb{G}^*$. The LRSW problem is hard if given (g, X, Y) and unlimited access to $\mathcal{O}(\cdot)$, there is no PPT algorithm that can solve for a new triple T^* for m^* which not queried to $\mathcal{O}(m^*)$ in polynomial time t with non-negligible advantage ϵ.

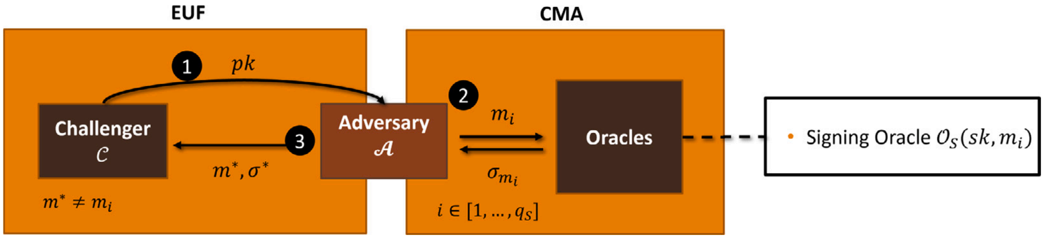

Fig. 1. An illustration of the EUF-CMA security model.

Definition 3 (*q-SDH Assumption [24]*)**.** The q-SDH problem is hard if given $(g, g^a, \ldots, g^{a^n})$, there is no PPT algorithm that can solve for $(w, g^{\frac{1}{a+w}})$ where $w \in \mathbb{Z}_p^*$ in polynomial time t with non-negligible advantage ϵ.

Definition 4 (*q-MSDH Assumption [23]*)**.** The q-Modified(M)SDH problem is hard if given $\{(g^{a^i}, g^{x \cdot a^i})\}_{i=0}^{q+1}$ and $(g^x, g^{x \cdot y \cdot a})$, there is no PPT algorithm that can solve for $(w, P, h^{\frac{1}{a+w}}, h^{\frac{x}{a \cdot P(a)}})$ for some $h \in \mathbb{G}$, P is a polynomial of degree at most q, and $w \neq 0$ such that $X + w$ and $P(X)$ are relatively prime, in polynomial time t with non-negligible advantage ϵ.

2.1. Definition of digital signatures

A signature scheme is defined based on the following three main algorithms.

- $\texttt{KeyGen}(1^k)$: On input security parameters 1^k. This key generation algorithm returns a public and secret key pair (pk, sk).
- $\texttt{Sign}(sk, m)$: On input secret key sk and message m. This signing algorithm returns a signature σ_m on m.
- $\texttt{Verify}(pk, m, \sigma_m)$: On input public key pk and message and signature pair (m, σ_m). This verification algorithm returns 1 or 0 which indicates a valid or invalid message and signature pair, respectively, corresponding to pk.

Correctness. Suppose a public and secret key pair $(pk, sk) \leftarrow \texttt{KeyGen}(1^k)$ is correctly generated. Given a signature on m signed with sk, such that $\sigma_m \leftarrow \texttt{Sign}(sk, m)$, the verification holds $\texttt{Verify}(pk, \sigma_m, m) = 1$.

2.2. The EUF-CMA security model

We recall the security model of existential unforgeability against chosen message attacks (EUF-CMA) [14]. An EUF-CMA secure signature scheme guarantees that it is infeasible for an adversary to forge a signature on a new message that has not been signed, even though the adversary obtains many signatures. The EUF-CMA security game is defined between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ as follows. Fig. 1 illustrates the game.

- **Setup Phase:** The challenger $\mathcal{C}$ first setups the environment by generating a public and secret key pair $(pk, sk) \leftarrow \texttt{KeyGen}$. $\mathcal{C}$ then forwards pk to $\mathcal{A}$.
- **Query Phase:** $\mathcal{A}$ may adaptively query for signatures on any message m_i. $\mathcal{C}$ returns signature σ_{m_i} on m_i, such that $\sigma_{m_i} \leftarrow \texttt{Sign}(sk, m_i)$.
- **Forgery Phase:** $\mathcal{A}$ returns a challenge message and signature pair (m^*, σ^*). $\mathcal{A}$ wins if the verification holds and m^* has not been queried before.

Definition 5 (*EUF-CMA*)**.** A signature scheme is (ϵ, q_s, t)-secure in EUF-CMA security model if there is no PPT adversary who runs in time t and makes at most q_s signing queries has the advantage at most ϵ in winning the game.

2.3. Security reduction

The security proof of a scheme usually describes a reduction that turns an adversary A who breaks the scheme with probability ϵ_A into an adversary B who breaks some underlying problems with probability $\epsilon_B \geq \frac{\epsilon_A}{L}$, where $L \geq 1$ indicates the loss factor. The reduction is tight if L is a small constant without any depending factors, e.g. the number of queries and the chance of successfully solving the problem made by the adversary during the game. A scheme with tight security reduction is meaningful as it can be implemented with optimal parameters in practice, hence achieving the scheme's efficiency as proposed without compensating the parameter size due to the security loss. For instance, suppose the underlying problem is $\epsilon_B = 2^{-128}$ hard and the reduction loss is $L = 2^{60}$, then the scheme can only preserve 68 bits security in practice $\epsilon_A = 2^{60} \cdot 2^{-128} = 2^{-68}$. This means that if we are deploying the 128 bits secure scheme, the security parameter must be at least 188 bits, which has degraded the scheme efficiency. As a result, reduction loss is an important factor as it directly affects the length of the security parameters needed, hence its efficiency, in practice [48–50].

2.4. The algebraic algorithms

The algebraic group model (AGM) is an idealized model that was formalized by Fuchsbauer, Kiltz, and Loss [19]. In the AGM, we assume the adversary's computation behaves algebraically. Suppose an algebraic adversary is given group elements $(X_1, \ldots, X_n) \in \mathbb{G}^n$. Whenever the adversary returns a group element $Z \in \mathbb{G}$, it also returns representation vector $\vec{\pi} = (\pi_1, \ldots, \pi_n) \in \mathbb{Z}_p^n$, such that $Z = \prod_{i=1}^n X_i^{\pi_i}$.

2.5. (Generalized) pedersen's commitment scheme

Pedersen's commitment scheme [51] allows the sender to securely commit a message while keeping it hidden from others. It also has the ability to reveal the committed message later. We recall its generalized version [52], which can commit multiple messages $(m_1, \ldots, m_n) \in \mathbb{Z}_p^n$, with the algorithms $\texttt{GP} = \{\texttt{Commit}, \texttt{Open}\}$ as follows. Let $g, h_1, \ldots, h_n \in \mathbb{G}^{n+1}$ be some group generators.

- $\texttt{GP.Commit}$. To commit messages $(m_1, \ldots, m_n) \in \mathbb{Z}_p^n$, the sender randomly selects $t \in \mathbb{Z}_p^*$ and sends commitment $C = g^t \cdot \prod_{i=1}^n h_i^{m_i}$ to the receiver.
- $\texttt{GP.Open}$. To open a commitment C, the sender reveals $(t, m_1, \ldots, m_n)$ to the receiver. The receiver accepts the opening if the verification holds such that $C = g^t \cdot \prod_{i=1}^n h_i^{m_i}$.

2.6. Sigma protocol on (generalized) pedersen's commitment

One can adopt the sigma protocol Σ.PoK for proving the knowledge of (Generalized) Pederson's commitment C without revealing messages to the receiver. It is run between a prover (sender) and a verifier as follows.

Σ.PoK: Suppose both of them have the common inputs, i.e., $g, h_1, \ldots, h_n \in \mathbb{G}^{n+1}$ and commitment $C = g^t \cdot \prod_{i=1}^n h_i^{m_i}$. The prover wishes to prove he knows $(t, m_1, \ldots, m_n)$ by initiating the following protocol.

- (P1): The prover randomly selects $t_0, t_1, \dots, t_n \in \mathbb{Z}_p^*$ and forwards $M = g^{t_0} \cdot \prod_{i=1}^{n} h_i^{t_i}$ to the verifier.
- (V2): The verifier replies a random $c \in \mathbb{Z}_p^*$ to the signer.
- (P3): The prover sets $s_0 = t_0 + c \cdot t$ and $\forall i \in [1, \dots, n] : s_i = t_i + c \cdot m_i$, and returns $(s_0, s_1, \dots, s_n)$ to the verifier.
- (V4): The verifier checks whether the following equation holds

  $$g^{s_0} \cdot \prod_{i=1}^{n} h_i^{s_i} = C^c \cdot M.$$

  The verification result is defined as $\pi = 1$ if it holds and $\pi = 0$ otherwise.

3. Our randomizable signatures

In this section, we propose a new randomizable signature scheme, namely CL+ signature scheme. For simplicity, we first give a single-message signature scheme, namely simpleCL+ signature scheme. We show that its security can be tightly reduced to the DLog problem under the EUF-CMA security model in the AGM. We then extend simpleCL+ signature scheme to cover a vector of message $\vec{m} = (m_1, \dots, m_n) \in \mathbb{Z}_p^n$, obtaining the CL+ signature scheme. Lastly, we show that the security of CL+ signature scheme is reducible to the security of simpleCL+ signature scheme.

3.1. SimpleCL+ signature scheme: Single-message signature scheme

Let $param = (p, g, e, \mathbb{G}, \mathbb{G}_t)$ be the system parameters, which is also the bilinear group tuple.

- KeyGen: On input security parameter 1^k, it randomly selects $x, y, z \in \mathbb{Z}_p^*$, computes $X = g^x, Y = g^y, Z = g^z$, and returns a public and secret key pair (pk, sk) such that

  $$pk = (X, Y, Z), \quad sk = (x, y, z).$$

- Sign: On input $(param, sk)$, and message $m \in \mathbb{Z}_p^*$. It randomly selects $R \in \mathbb{G}^*$, such that $R \neq 1$, and returns signature $\sigma_m = (\sigma_m^{(1)}, \sigma_m^{(2)}, \sigma_m^{(3)})$ on m as follows

  $$\sigma_m^{(1)} = R,$$
  $$\sigma_m^{(2)} = (\sigma_m^{(1)})^x,$$
  $$\sigma_m^{(3)} = (\sigma_m^{(2)})^{y+mz}.$$

- Verify: On input $(param, pk, m, \sigma_m)$, it returns 1, which indicates a valid signature on m if both of the following equations hold, such that

  $$e(\sigma_m^{(2)}, g) = e(\sigma_m^{(1)}, X), \quad \text{and}$$
  $$e(\sigma_m^{(3)}, g) = e(\sigma_m^{(2)}, YZ^m)$$

  Otherwise, it returns 0.

Correctness. If signature $\sigma_m = (\sigma_m^{(1)}, \sigma_m^{(2)}, \sigma_m^{(3)}) = (R, R^x, (R^x)^{(y+mz)})$ on m generated with user secret key x, y, z, then the verification algorithm holds, such that

$$\begin{aligned} e(\sigma_m^{(2)}, g) &= e(R^x, g) = e(R, g^x) = e(\sigma_m^{(1)}, X), \quad \text{and} \\ e(\sigma_m^{(3)}, g) &= e((\sigma_m^{(2)})^{(y+mz)}, g) = e(\sigma_m^{(2)}, g^{y+mz}) \\ &= e(\sigma_m^{(2)}, YZ^m). \end{aligned}$$

Randomizability. Given signature $\sigma_m = (\sigma_m^{(1)}, \sigma_m^{(2)}, \sigma_m^{(3)})$ on m. Let $r \in \mathbb{Z}_p^*$ be some randomness, such that

$$\sigma_m^{(1)} = R = g^r, \quad \sigma_m^{(2)} = g^{xr}, \quad \sigma_m^{(3)} = g^{xr(y+mz)}.$$

The signature is randomizable as setting $(\sigma_m)^{r'} = ((\sigma_m^{(1)})^{r'}, (\sigma_m^{(2)})^{r'}, (\sigma_m^{(3)})^{r'})$, where $r' \in \mathbb{Z}_p^*$ is some freshly chosen randomness, the signature is valid, such that new randomness becomes $\hat{r} = r \cdot r'$. The fresh signature is valid, where

$$(\sigma_m^{(1)})^{r'} = g^{\hat{r}}, \quad (\sigma_m^{(2)})^{r'} = g^{x\hat{r}}, \quad (\sigma_m^{(3)})^{r'} = g^{x\hat{r}(y+mz)}.$$

3.1.1. Security analysis of SimpleCL+ signature scheme

We first elaborate on the security proof at a high level. We prove the scheme is secure under the DLog assumption and EUF-CMA in the AGM. Given a DLog problem instance g^a, we can program three indistinguishable three simulations $\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{R}_3$ that embed g^a into public key X, or every signature randomness $\sigma_{m_i}^{(1)} = R_i$, or public key Y, Z, respectively. The simulations are summarized in Table 1 where $x_0, y_0, z_0, y_1, z_1, r_i, u_i, v_i \in \mathbb{Z}_p^*$ are randomly chosen. Note that all signature queries can be answered without aborts. After obtaining the forgery and corresponding representations, we use the relation $\sigma_{m^*}^{(3)} = (\sigma_{m^*}^{(1)})^{x(y+m^*z)}$ and $\sigma_{m^*}^{(1)} \neq 1$ to derive a modular equation, which enables the simulator to solve for the DLog solution under certain adversary behaviors **E**, **F** and the corresponding simulations. In particular, the simulation $\mathcal{R}_1$ succeeds if **E** does not hold; $\mathcal{R}_2$ succeeds if **E** holds and **F** does not hold; and $\mathcal{R}_3$ succeeds if both **E** and **F** hold. Fig. 2 illustrates the high-level idea of security proof.

Theorem 1. *If there exists an adversary who can break the EUF-CMA security of the simpleCL+ signature scheme in the AGM, then there exists an adversary that can solve DLog problem with a success probability of $\frac{1}{3}$.*

Proof of Theorem 1. Suppose there exists an algebraic adversary who can (t, q_s, ϵ)-break Scheme A under EUF-CMA in the AGM. Given as input a DLog problem instance (g, g^a) over the pairing group tuple $(p, g, e, \mathbb{G}, \mathbb{G}_T)$. We can design three indistinguishable simulations $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3$ that can run either one of them to solve the DLog problem with the forgery given by the algebraic adversary.

Simulation $\mathcal{R}_1$. The first simulation $\mathcal{R}_1$ is programmed as follows. The simulator uses the problem instance as the public key $X = g^x = g^a$.

Setup. The simulator randomly select $y_0, z_0 \in \mathbb{Z}_p^*$ and sets

$$X = g^a, \quad Y = g^{y_0}, \quad Z = g^{z_0}.$$

The public key $pk = (X, Y, Z)$ is returned.

Signing Query $\mathcal{O}_S(m_i)$. The adversary makes signing queries in this phase. The simulator randomly selects $r_i \in \mathbb{Z}_p^*$ and sets signature $\sigma_{m_i} = (\sigma_{m_i}^{(1)}, \sigma_{m_i}^{(2)}, \sigma_{m_i}^{(3)})$ as follows.

$$\sigma_{m_i}^{(1)} = g^{r_i},$$
$$\sigma_{m_i}^{(2)} = g^{ar_i},$$
$$\sigma_{m_i}^{(3)} = g^{ar_i(y_0+m_iz_0)}$$

Correctness. It is easy to see σ_{m_i} is a valid signature, such that

$$\begin{aligned} \sigma_{m_i}^{(3)} &= g^{ar_i(y_0+m_iz_0)} \\ &= (X^{r_i})^{y_0+m_iz_0} \\ &= (\sigma_{m_i}^{(2)})^{y+m_iz} \end{aligned}$$

Forgery. The algebraic adversary returns a forged signature $\sigma_{m^*} = (\sigma_{m^*}^{(1)}, \sigma_{m^*}^{(2)}, \sigma_{m^*}^{(3)})$ on a chosen message m^*, which has not been queried to for signature. The simulator receives the forgery as follows, such that for some $r^* \in \mathbb{Z}_p^*$,

$$\sigma_{m^*}^{(1)} = g^{r^*},$$
$$\sigma_{m^*}^{(2)} = (\sigma_{m^*}^{(1)})^x,$$
$$\sigma_{m^*}^{(3)} = (\sigma_{m^*}^{(2)})^{y+m^*z},$$

and their respective representation vectors $\vec{\alpha}, \vec{\beta}, \vec{\gamma}$, which is described as follows. To simplify the notation, let $\pi \in [\alpha, \beta, \gamma]$, and

$$\vec{\pi} = \pi_0, \pi_1, \pi_2, \pi_3, \pi_{4,1}, \dots, \pi_{4,q_s}, \pi_{5,1}, \dots, \pi_{5,q_s}, \pi_{6,1}, \dots, \pi_{6,q_s},$$

where q_s is the number of signing queries. The forged signature σ_{m^*} can be algebraically described based on all received group

Table 1
Simulation of SimpleCL+Signature scheme.

Element	Simulation $\mathcal{R}_1$	Simulation $\mathcal{R}_2$	Simulation $\mathcal{R}_3$
X	g^{a}	g^{x_0}	g^{x_0}
Y	g^{y_0}	g^{y_0}	$g^{ay_1+y_0}$
Z	g^{z_0}	g^{z_0}	$g^{az_1+z_0}$
$\sigma_{m_i}^{(1)}$	g^{r_i}	$g^{au_i+v_i}$	g^{r_i}
$\sigma_{m_i}^{(2)}$	g^{ar_i}	$g^{au_i x_0} g^{v_i x_0}$	$g^{x_0 r_i}$
$\sigma_{m_i}^{(3)}$	$g^{ar_i(y_0+m_i z_0)}$	$g^{au_i x_0(y_1+m_i z_1)} g^{v_i x_0(y_0+m_i z_0)}$	$g^{a(y_0+m_i z_0)x_0 r_i} g^{x_0 r_i(y_1+m_i z_1)}$

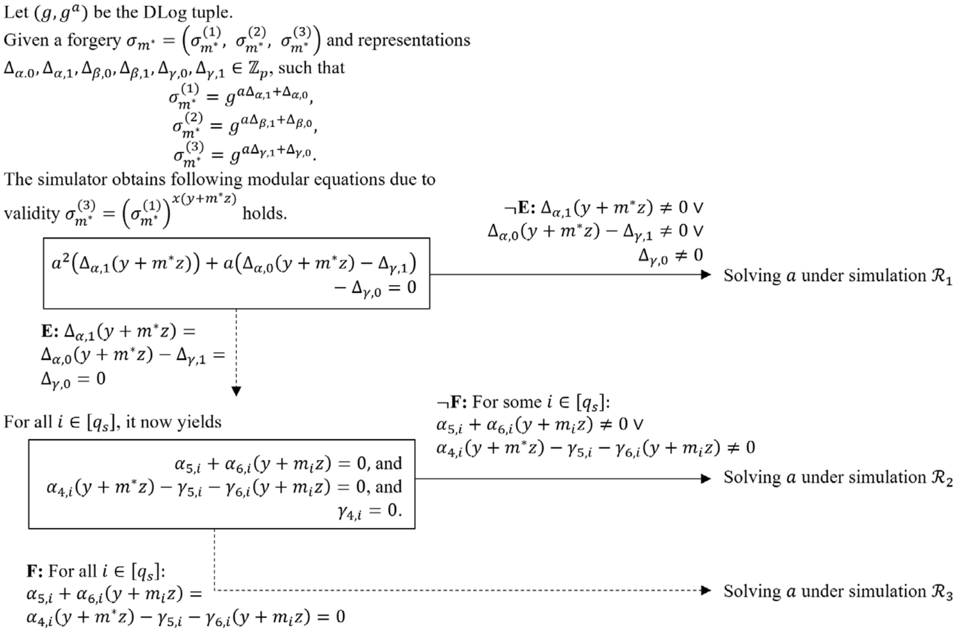
Let (g, g^a) be the DLog tuple.
Given a forgery $\sigma_{m^*} = \left(\sigma_{m^*}^{(1)}, \sigma_{m^*}^{(2)}, \sigma_{m^*}^{(3)}\right)$ and representations
$\Delta_{\alpha.0}, \Delta_{\alpha,1}, \Delta_{\beta,0}, \Delta_{\beta,1}, \Delta_{\gamma,0}, \Delta_{\gamma,1} \in \mathbb{Z}_p$, such that
$\sigma_{m^*}^{(1)} = g^{a\Delta_{\alpha,1}+\Delta_{\alpha,0}}$,
$\sigma_{m^*}^{(2)} = g^{a\Delta_{\beta,1}+\Delta_{\beta,0}}$,
$\sigma_{m^*}^{(3)} = g^{a\Delta_{\gamma,1}+\Delta_{\gamma,0}}$.
The simulator obtains following modular equations due to validity $\sigma_{m^*}^{(3)} = \left(\sigma_{m^*}^{(1)}\right)^{x(y+m^*z)}$ holds.
$a^2\left(\Delta_{\alpha,1}(y+m^*z)\right) + a\left(\Delta_{\alpha,0}(y+m^*z) - \Delta_{\gamma,1}\right) - \Delta_{\gamma,0} = 0$
$\neg$**E**: $\Delta_{\alpha,1}(y+m^*z) \neq 0 \vee$ $\Delta_{\alpha,0}(y+m^*z) - \Delta_{\gamma,1} \neq 0 \vee$ $\Delta_{\gamma,0} \neq 0$
Solving a under simulation $\mathcal{R}_1$
E: $\Delta_{\alpha,1}(y+m^*z) =$ $\Delta_{\alpha,0}(y+m^*z) - \Delta_{\gamma,1} =$ $\Delta_{\gamma,0} = 0$
For all $i \in [q_s]$, it now yields
$\alpha_{5,i} + \alpha_{6,i}(y+m_i z) = 0$, and $\alpha_{4,i}(y+m^*z) - \gamma_{5,i} - \gamma_{6,i}(y+m_i z) = 0$, and $\gamma_{4,i} = 0$.
$\neg$**F**: For some $i \in [q_s]$: $\alpha_{5,i} + \alpha_{6,i}(y+m_i z) \neq 0 \vee$ $\alpha_{4,i}(y+m^*z) - \gamma_{5,i} - \gamma_{6,i}(y+m_i z) \neq 0$
Solving a under simulation $\mathcal{R}_2$
F: For all $i \in [q_s]$: $\alpha_{5,i} + \alpha_{6,i}(y+m_i z) =$ $\alpha_{4,i}(y+m^*z) - \gamma_{5,i} - \gamma_{6,i}(y+m_i z) = 0$
Solving a under simulation $\mathcal{R}_3$

Fig. 2. An overview of the security reduction.

elements in the following general form. For $n \in [1, 2, 3]$ and $\pi \in [\alpha, \beta, \gamma]$, each $\sigma_{m^*} = (\sigma_{m^*}^{(1)}, \sigma_{m^*}^{(2)}, \sigma_{m^*}^{(3)})$ is defined as

$$\sigma_{m^*}^{(n)} = g^{\pi_0} X^{\pi_1} Y^{\pi_2} Z^{\pi_3} \prod_{i=1}^{q_s} (\sigma_{m_i}^{(1)})^{\pi_{4,i}} (\sigma_{m_i}^{(2)})^{\pi_{5,i}} (\sigma_{m_i}^{(3)})^{\pi_{6,i}}$$
$$= g^{\pi_0} g^{x\pi_1} g^{y\pi_2} g^{z\pi_3} \prod_{i=1}^{q_s} (g^{r_i})^{\pi_{4,i}} \cdot (g^{xr_i})^{\pi_{5,i}} \cdot (g^{xr_i(y+m_i z)})^{\pi_{6,i}}.$$

Based on the definition of simulation, where $X = g^a$ is set, the above equation can be written as

$$\sigma_{m^*}^{(n)} = (g^a)^{\pi_1 + \sum_{i=1}^{q_s} r_i \pi_{5,i} + r_i \pi_{6,i}(y+m_i z)} \cdot g^{\pi_0 + y\pi_2 + z\pi_3 + \sum_{i=1}^{q_s} r_i \pi_{4,i}}$$
$$= g^{a\Delta_{\pi,1} + \Delta_{\pi,0}},$$

such that

$$\Delta_{\pi,1} = \pi_1 + \sum_{i=1}^{q_s} r_i \pi_{5,i} + r_i \pi_{6,i}(y + m_i z),$$

$$\Delta_{\pi,0} = \pi_0 + y\pi_2 + z\pi_3 + \sum_{i=1}^{q_s} r_i \pi_{4,i}.$$

Therefore, let $DLog(\cdot)$ denote the DLog of some group element, the forgery and representations yield the following three equations

$$DLog(\sigma_{m^*}^{(1)}) = a\Delta_{\alpha,1} + \Delta_{\alpha,0},$$
$$DLog(\sigma_{m^*}^{(2)}) = a\Delta_{\beta,1} + \Delta_{\beta,0},$$
$$DLog(\sigma_{m^*}^{(3)}) = a\Delta_{\gamma,1} + \Delta_{\gamma,0},$$

where $a \in \mathbb{Z}_p^*$ is the unknown DLog solution, and $\Delta_{\alpha,0}, \Delta_{\alpha,1}, \Delta_{\beta,0}, \Delta_{\beta,1}, \Delta_{\gamma,0}, \Delta_{\gamma,1}$ are coefficients based on given representations $\vec{\alpha}, \vec{\beta}, \vec{\gamma}$ and known m_i , m^*, y_0, z_0, r_i. Suppose the forged signature is valid. This implies that the equation $\sigma_{m^*}^{(3)} = (\sigma_{m^*}^{(1)})^{x(y+m^*z)}$ and $(\sigma_{m^*}^{(1)}) \neq 1$ must hold, and it yields

$$DLog(\sigma_{m^*}^{(3)}) = a\Delta_{\gamma,1} + \Delta_{\gamma,0}$$
$$= DLog(\sigma_{m^*}^{(1)})(x(y + m^*z))$$
$$= (a\Delta_{\alpha,1} + \Delta_{\alpha,0})(a(y + m^*z))$$

By rearranging the above terms, we can obtain the following modular quadratic equation

$$a^2(\Delta_{\alpha,1}(y + m^*z)) + a(\Delta_{\alpha,0}(y + m^*z) - \Delta_{\gamma,1}) - \Delta_{\gamma,0} = 0.$$

Aborts 1. The simulator aborts if adversary behavior **E** holds, such that all coefficients are zero, i.e., $\Delta_{\alpha,1}(y+m^*z) = \Delta_{\alpha,0}(y + m^*z) - \Delta_{\gamma,1} = \Delta_{\gamma,0} = 0$.

Otherwise, the simulator solves for at most two possible solutions a_0, a_1, and checks for the correct DLog solution $g^a = g^{a_0}$ or $g^a = g^{a_1}$.

Success Probability of $\mathcal{R}_1$. The simulation succeeds if behavior **E** does not occur. Hence, the success probability of simulation $\mathcal{R}_1$ is defined as follows.

$$\Pr[\mathcal{R}_1] = \Pr[\neg\mathbf{E}] = \frac{1}{2}$$

Simulation $\mathcal{R}_2$. The second simulation $\mathcal{R}_2$ is programmed as follows. The simulator embeds g^a into every randomness $\sigma_{m_i}^{(1)} = g^{r_i} = g^{au_i+v_i}$.

Setup. The simulator randomly select $x_0, y_0, z_0 \in \mathbb{Z}_p^*$ and sets

$$X = g^{x_0}, \quad Y = g^{y_0}, \quad Z = g^{z_0}.$$

The public key $pk = (X, Y, Z)$ is returned.

Signing Query $\mathcal{O}_S(m_i)$. The adversary makes signing queries in this phase. The simulator randomly selects $u_i, v_i \in \mathbb{Z}_p^*$ and sets signature $\sigma_{m_i} = (\sigma_{m_i}^{(1)}, \sigma_{m_i}^{(2)}, \sigma_{m_i}^{(3)})$ as follows.

$$\sigma_{m_i}^{(1)} = g^{au_i+v_i},$$
$$\sigma_{m_i}^{(2)} = g^{au_i x_0 + x_0 v_i},$$
$$\sigma_{m_i}^{(3)} = g^{au_i x_0(y_0+m_i z_0)+v_i x_0(y_0+m_i z_0)}$$

Correctness. It is easy to see σ_{m_i} is a valid signature, such that

$$\begin{aligned}\sigma_{m_i}^{(3)} &= g^{au_i x_0(y_0+m_i z_0)+v_i x_0(y_0+m_i z_0)}\\ &= (X^{au_i+v_i})^{y_0+m_i z_0}\\ &= (\sigma_{m_i}^{(2)})^{y+m_i z}\end{aligned}$$

Forgery. The algebraic adversary returns a forged signature $\sigma_{m^*} = (\sigma_{m^*}^{(1)}, \sigma_{m^*}^{(2)}, \sigma_{m^*}^{(3)})$ on a chosen message m^* and their respective representation vectors $\vec{\alpha}, \vec{\beta}, \vec{\gamma}$ following the definition as in $\mathcal{R}_1$. Let $T_{\alpha,1} = X^{\alpha_1} \cdot \prod(\sigma_{m_i}^{(2)})^{\alpha_{5,i}} \cdot (\sigma_{m_i}^{(3)})^{\alpha_{6,i}}$, $T_{\alpha,0} = g^{\alpha_0} \cdot Y^{\alpha_2} \cdot Z^{\alpha_3} \cdot \prod(\sigma_{m_i}^{(1)})^{\alpha_{4,i}}$, $T_{\gamma,1} = X^{\gamma_1} \cdot \prod(\sigma_{m_i}^{(2)})^{\gamma_{5,i}} \cdot (\sigma_{m_i}^{(3)})^{\gamma_{6,i}}$, and $T_{\gamma,0} = g^{\gamma_0} \cdot Y^{\gamma_2} \cdot Z^{\gamma_3} \prod(\sigma_{m_i}^{(1)})^{\gamma_{4,i}}$.

Abort 2. The simulator aborts if all the following equations hold

$$e(T_{\alpha,1}, YZ^{m^*}) \neq e(X, g),$$
$$e(T_{\alpha,0}, YZ^{m^*}) \neq e(T_{\gamma,1}, g), \text{ and}$$
$$e(T_{\gamma,0}, g) \neq 1.$$

Otherwise, it indicates the adversary behavior **E** holds and the simulator obtains the following equations, such that

$$\Delta_{\alpha,1}(y + m^* z) = \Delta_{\alpha,0}(y + m^* z) - \Delta_{\gamma,1} = \Delta_{\gamma,0} = 0,$$

where $\Delta_{\alpha,1} = \alpha_1 + \sum_{i=1}^{q_s} r_i \alpha_{5,i} + r_i \alpha_{6,i}(y + m_i z)$, $\Delta_{\alpha,0} = \alpha_0 + y\alpha_2 + z\alpha_3 + \sum_{i=1}^{q_s} r_i \gamma_{4,i}$, $\Delta_{\gamma,1} = \gamma_1 + \sum_{i=1}^{q_s} r_i \gamma_{5,i} + r_i \gamma_{6,i}(y + m_i z)$, and $\Delta_{\gamma,0} = \gamma_0 + y\gamma_2 + z\gamma_3 + \sum_{i=1}^{q_s} r_i \gamma_{4,i} = 0$. Besides, the simulator aborts if adversary behavior **F** holds, such that, for all $i \in [q_s]$,

$$\alpha_{5,i} + \alpha_{6,i}(y + m_i z) = 0, \text{ and}$$
$$\alpha_{4,i}(y + m^* z) - \gamma_{5,i} - \gamma_{6,i}(y + m_i z) = 0, \text{ and}$$
$$\gamma_{4,i} = 0.$$

Otherwise, the simulator solves for DLog solution in either of the following cases.

Case A1. $\Delta_{\alpha,1}(y+m^* z) = 0$. By rearranging with the term r_i, we obtain

$$\sum_{i=1}^{q_s} r_i(\alpha_{5,i} + \alpha_{6,i}(y + m_i z)) = -\alpha_1.$$

Due to definition of simulation $\mathcal{R}_2$, we can solve for a if there exists at least one non-zero coefficient, i.e. $\exists i \in [q_s] : \alpha_{5,i} + \alpha_{6,i}(y + m_i z) \neq 0$, such that

$$a = \frac{-\alpha_1 - \sum v_i(\alpha_{5,i} + \alpha_{6,i}(y_0 + m_i z_0))}{\sum u_i(\alpha_{5,i} + \alpha_{6,i}(y_0 + m_i z_0))}.$$

Case A2. $\Delta_{\alpha,0}(y+m^* z) - \Delta_{\gamma,1} = 0$. By rearranging with the term r_i, we obtain

$$\begin{aligned}&\sum_{i=1}^{q_s} r_i(\alpha_{4,i}(y + m^* z) - \gamma_{5,i} - \gamma_{6,i}(y + m_i z))\\ &= \gamma_1 - (y + m^* z)(\alpha_0 + y\alpha_2 + z\alpha_3).\end{aligned}$$

Let $\omega_i = \alpha_{4,i}(y + m^* z) - \gamma_{5,i} - \gamma_{6,i}(y + m_i z)$. Due to the definition of simulation $\mathcal{R}_2$, we can solve for a if there exists at least one non-zero coefficient. For some $i \in [q_s] : \omega_i \neq 0$,

$$a = \frac{\gamma_1 - (y_0 + m^* z_0)(\alpha_0 + y_0\alpha_2 + z_0\alpha_3) - \sum v_i(\omega_i)}{\sum u_i(\omega_i)}.$$

Case A3. $\Delta_{\gamma,0} = \gamma_0 + y\gamma_2 + z\gamma_3 + \sum_{i=1}^{q_s} r_i \gamma_{4,i} = 0$. By rearranging with the term r_i, we obtain

$$\sum_{i=1}^{q_s} r_i \gamma_{4,i} = -\gamma_0 - y\gamma_2 - z\gamma_3.$$

Due to definition of simulation $\mathcal{R}_2$, we can solve for a as long as there exists at least one non-zero coefficient $\exists i \in [q_s] : \gamma_{4,i} \neq 0$, such that

$$a = \frac{-\gamma_0 - y_0\gamma_2 - z_0\gamma_3 - \sum v_i(\gamma_{4,i})}{\sum u_i(\gamma_{4,i})}.$$

Success Probability of $\mathcal{R}_2$. The simulation succeeds if behavior **E** holds and **F** does not occur. Hence, the success probability of simulation $\mathcal{R}_2$ is defined as follows.

$$\Pr[\mathcal{R}_2] = \Pr[\mathbf{E}] \cdot \Pr[\neg\mathbf{F}] = \frac{1}{4}$$

Simulation $\mathcal{R}_3$. The third simulation $\mathcal{R}_3$ is programmed as follows. The simulator embeds g^a into public key $Y = g^y = g^{ay_1+y_0}$ and $Z = g^z = g^{az_1+z_0}$.

Setup. The simulator randomly select $x_0, y_0, y_1, z_0, z_1 \in \mathbb{Z}_p^*$ and sets

$$X = g^{x_0}, \quad Y = g^{ay_1+y_0}, \quad Z = g^{az_1+z_0}.$$

The public key $pk = (X, Y, Z)$ is returned.

Signing Query $\mathcal{O}_S(m_i)$. The adversary makes signing queries in this phase. The simulator randomly selects $r_i \in \mathbb{Z}_p^*$ and sets signature $\sigma_{m_i} = (\sigma_{m_i}^{(1)}, \sigma_{m_i}^{(2)}, \sigma_{m_i}^{(3)})$ as follows.

$$\sigma_{m_i}^{(1)} = g^{r_i},$$
$$\sigma_{m_i}^{(2)} = g^{xr_i},$$
$$\sigma_{m_i}^{(3)} = g^{axr_i(y_1+m_i z_1)+xr_i(y_0+m_i z_0)}$$

Correctness. It is easy to see σ_{m_i} is a valid signature, such that

$$\begin{aligned}\sigma_{m_i}^{(3)} &= g^{axr_i(y_1+m_i z_1)+xr_i(y_0+m_i z_0)}\\ &= (X^{r_i})^{a(y_1+m_i z_1)+y_0+m_i z_0}\\ &= (\sigma_{m_i}^{(2)})^{y+m_i z}\end{aligned}$$

Forgery. The algebraic adversary returns a forged signature $\sigma_{m^*} = (\sigma_{m^*}^{(1)}, \sigma_{m^*}^{(2)}, \sigma_{m^*}^{(3)})$ on a chosen message m^* and their respective representation vectors $\vec{\alpha}, \vec{\beta}, \vec{\gamma}$ following the definition as in $\mathcal{R}_2$.

Abort 3. The simulator aborts following the similar definition as in **Abort 2**, indicating behavior **E** holds. However, the simulator checks the following condition, adversary behavior **F**, and aborts if all the following equations hold. For all $i \in [q_s]$,

$$g^{\alpha_{5,i}} \cdot (YZ^{m^*})^{\alpha_{6,i}} \neq 1,$$
$$(YZ^{m^*})^{\alpha_{4,i}} \neq g^{\gamma_{5,i}}(YZ^{m^*})^{\gamma_{6,i}}, \text{ and}$$
$$\gamma_{4,i} \neq 0.$$

The simulator obtains equations as follows, such that for all $i \in [q_s]$,

$$\alpha_{5,i} + \alpha_{6,i}(y + m_i z) = 0, \text{ and}$$
$$\alpha_{4,i}(y + m^* z) - \gamma_{5,i} - \gamma_{6,i}(y + m_i z) = 0, \text{ and}$$

$\gamma_{4,i} = 0$.

The simulator solves for DLog solution in either of the following cases.

Case B1. $\forall i : \alpha_{5,i}+\alpha_{6,i}(y+m_i z) = 0$. Note that it also implies that $\alpha_1 = 0$ because of **Case A1**, such that $\sum_{i=1}^{q_s} r_i(\alpha_{5,i}+\alpha_{6,i}(y+m_i z)) = -\alpha_1$. Due to definition of simulation $\mathcal{R}_3$, we can solve for a

$$a = \frac{-\alpha_{5,i} - \alpha_{6,i}(y_0 + m_i z_0)}{\alpha_{6,i}(y_1 + m_i z_1)}$$

as long as there exists at least one non-zero coefficients $\alpha_{6,i} \neq 0$. Otherwise, it implies the condition that $\forall i \in [q_s] : \alpha_{5,i} = \alpha_{6,i} = 0$ must hold.

Case B2. $\forall i : \alpha_{4,i}(y + m^* z) - \gamma_{5,i} - \gamma_{6,i}(y + m_i z) = 0$. Due to **Case A2**, it also yields equation

$$\gamma_1 = (y + m^* z)(\alpha_0 + y\alpha_2 + z\alpha_3).$$

Based on the definition of $\mathcal{R}_3$, if either $\alpha_0, \alpha_2, \alpha_3$ is non-zero in $\mathbb{Z}_p^*$, it is easy to see the above equation yields a modular quadratic equation that allows us to solve for at most two possible a. Hence, there is no need to proceed with the remaining steps. Otherwise, it is worth noting that the above equation implies that $\alpha_0 = \alpha_2 = \alpha_3 = 0$ holds. Therefore, it remains to analyze for all $i \in [q_s]$ that

$$\alpha_{4,i}(y + m^* z) - \gamma_{5,i} - \gamma_{6,i}(y + m_i z) = 0$$

Again, due to definition of $\mathcal{R}_3$, we can solve for a by computing

$$a = \frac{\gamma_{5,i} + \gamma_{6,i}(y_0 + m_i z_0) - \alpha_{4,i}(y_0 + m^* z_0)}{\alpha_{4,i}(y_1 + m^* z_1) - \gamma_{6,i}(y_1 + m_i z_1)}.$$

Note that adversary behavior **E** and **F** guarantee that $\alpha_0 = \alpha_1 = \alpha_2 = \alpha_3 = 0$ must hold. In addition, if the simulator cannot solve for the DLog solution under **Case B1**, it implies that $\forall i \in [q_s] : \alpha_{5,i} = \alpha_{6,i} = 0$. Therefore, we have at least one non-zero $\exists i \in [q_s] : \alpha_{4,i} \neq 0$ must hold due to signature definition, i.e. $\sigma_{m^*}^{(1)} = g^{\sum_i^{q_s} \alpha_{4,i} r_i} \neq 1$.

It remains to argue in the case that there exists condition $\alpha_{4,i} \neq 0$ and $\alpha_{4,i}(y_1 + m^* z_1) - \gamma_{6,i}(y_1 + m_i z_1) = 0$ hold. Fortunately, this cannot be true because if there exists $\alpha_{4,i} = \gamma_{6,i}$, it implies that the forged message is identical to one of the queried message, i.e. $m^* = m_i$, which contradicts the EUF-CMA security game. While $\alpha_{4,i} \neq \gamma_{6,i}$, the adversary has no advantage to sets $m^* = \frac{\gamma_{6,i}(y_1+m_i z_1)-\alpha_{4,i} y_1}{\alpha_{4,i} z_1}$ because $y_1, z_1 \in \mathbb{Z}_p^*$ are both information-theoretically hidden from the view of the adversary.

Success Probability of $\mathcal{R}_3$. The simulation succeeds if both behavior **E** and **F** hold. Hence, the success probability of simulation $\mathcal{R}_3$ is defined as follows.

$$\Pr[\mathcal{R}_3] = \Pr[\mathbf{E}] \cdot \Pr[\mathbf{F}] = \frac{1}{4}$$

Indistinguishable Simulations. The three simulations are indistinguishable as all elements look random from the view of the adversary. The correctness of the simulations holds for every signature query.

The Concrete Security. The security loss is three, which can be obtained based on the final success probability of the reduction. Suppose the simulator randomly runs one of the three simulations. It is easy to see the simulator can successfully extract the DLog solution with at least $\frac{1}{3}$ of the success probability as follows.

$$\begin{aligned}\Pr[\text{Success}] &= \frac{1}{3} \cdot \Pr[\mathcal{R}_1] + \frac{1}{3} \cdot \Pr[\mathcal{R}_2] + \frac{1}{3} \cdot \Pr[\mathcal{R}_3] \\ &= \frac{1}{3} \cdot (\frac{1}{2} + \frac{1}{4} + \frac{1}{4}) \\ &= \frac{1}{3}.\end{aligned}$$

3.2. The CL+ signature scheme: Multi-messages signature scheme

We are now ready to extend the simpleCL+ signature scheme to sign multiple messages at once, namely the CL+ signature scheme. The algorithms of this scheme are identical to the single-message version. It is worth noting that we define signing message becomes a n−message vector, i.e., $\vec{m} = (m_1, \dots, m_n) \in \mathbb{Z}_p^n$.

- `KeyGen`: On input security parameter 1^k, it randomly selects $x, y, z_1, \dots, z_n \in \mathbb{Z}_p^*$, computes $X = g^x, Y = g^y, Z_1 = g^{z_1}, \dots, Z_n = g^{z_n}$, and returns a public and secret key pair (pk, sk) such that

  $$pk = (X, Y, Z_1, \dots, Z_n), \quad sk = (x, y, z_1, \dots, z_n).$$

- `Sign`: On input $(param, sk)$, and message $\vec{m} = (m_1, \dots, m_n) \in \mathbb{Z}_p^n$. It randomly selects $R \in \mathbb{G}^*$ and returns signature $\sigma_m = (\sigma_{\vec{m}}^{(1)}, \sigma_{\vec{m}}^{(2)}, \sigma_{\vec{m}}^{(3)})$ on $\vec{m}$ as follows

  $$\sigma_{\vec{m}}^{(1)} = R,$$
  $$\sigma_{\vec{m}}^{(2)} = (\sigma_{\vec{m}}^{(1)})^x,$$
  $$\sigma_{\vec{m}}^{(3)} = (\sigma_{\vec{m}}^{(2)})^{y+\sum_{i=1}^n m_i z_i}.$$

- `Verify`: On input $(param, pk, \vec{m}, \sigma_m)$, it returns 1, which indicates a valid signature on m if both of the following equations hold, such that

  $$e(\sigma_{\vec{m}}^{(2)}, g) = e(\sigma_{\vec{m}}^{(1)}, X), \quad \text{and}$$
  $$e(\sigma_{\vec{m}}^{(3)}, g) = e(\sigma_{\vec{m}}^{(2)}, Y \prod_{i=1}^n Z_i^{m_i})$$

  Otherwise, it returns 0.

Correctness. Given a signature $\sigma_m = (\sigma_{\vec{m}}^{(1)}, \sigma_{\vec{m}}^{(2)}, \sigma_{\vec{m}}^{(3)})$ on m generated with user secret key x, y, z. Its validity holds, such that

$$e(\sigma_{\vec{m}}^{(2)}, g) = e(R^x, g) = e(R, g^x) = e(\sigma_{\vec{m}}^{(1)}, X), \quad \text{and}$$

$$\begin{aligned}e(\sigma_{\vec{m}}^{(3)}, g) &= e((\sigma_{\vec{m}}^{(2)})^{y+\sum_{i=1}^n m_i z_i}, g) \\ &= e(\sigma_{\vec{m}}^{(2)}, g^{y+\sum_{i=1}^n m_i z_i}) \\ &= e(\sigma_{\vec{m}}^{(2)}, Y \prod_{i=1}^n Z_i^{m_i})\end{aligned}$$

Randomizability. Since the signature structure follows the same as in simpleCL+ signature scheme, it is easy to see the randomizability holds.

3.2.1. Security analysis of CL+ signature scheme

The EUF-CMA security of CL+ signature scheme is based on the security of simpleCL+ signature scheme.

Theorem 2. *If there exists an adversary that can break the EUF-CMA security of CL+ signature scheme, then there is an adversary who can break the EUF-CMA security of simpleCL+ signature scheme.*

Proof of Theorem 2 (*Proof Sketch*)**.** We only provide a proof sketch as the full proof follows an identical technique as in [20], such that we construct a simulator $\mathcal{S}$ that makes use of the adversary $\mathcal{A}$ against the EUF-CMA security of CL+ signature scheme to break the EUF-CMA security of simpleCL+ signature scheme.

Given a challenge public key $\hat{pk} = (\hat{X}, \hat{Y}, \hat{Z})$ and an accessible signing oracle $\mathcal{O}_S$ from the EUF-CMA security game of simpleCL+ signature scheme, the simulator $\mathcal{S}$ extends $\hat{Z}$ into $Z_1, \dots, Z_n \in \mathbb{G}^n$ of public key elements, such that for $i = 1, \dots, n$, $\mathcal{S}$ selects $\mu_i, \nu_i \in \mathbb{Z}_p$ and sets $Z_i = (\hat{Z})^{\mu_i} g^{\nu_i}$. The adversary $\mathcal{A}$ of CL+ signature scheme is given public key $pk = (X = \hat{X}, Y = \hat{Y}, Z_1, \dots, Z_n)$.

During the query phase, $\mathcal{A}$ may query any signature on vector of messages $\vec{m_i} = (m_{i,1}, \dots, m_{i,n})$ to $\mathcal{S}$. In this case, $\mathcal{S}$ aggregates the vector into a single message by computing $m_i = \sum_{j=1}^n m_{i,j} \mu_j$ and obtaining

Table 2
A Comparison of CL-like Signatures under EUF-CMA. Here, n refers to the number of messages to be signed, i.e., $m = (m_1, \ldots, m_n) \in \mathbb{Z}_p^*$.

	Signature size	Assumption	Rand.	AGM
CL [1,19]	$(1+2n) \times \mathbb{G}$	LRSW DLog	Full Full	– Loose
Modified CL [23]	$(3+2n) \times \mathbb{G}+$ $1 \times \mathbb{Z}_p$	q-MSDH	Weak	–
This work, CL+	$3 \times \mathbb{G}$	DLog	Full	Tight

signature via signing oracle $\mathcal{O}_S(m_i) \rightarrow \sigma_{m_i} = (\sigma_{m_i}^{(1)}, \sigma_{m_i}^{(2)}, \sigma_{m_i}^{(3)})$. The signature is valid on $(\hat{pk}, m_i)$, such that the following equations hold,

$$e(\sigma_{m_i}^{(2)}, g) = e(\sigma_{m_i}^{(1)}, \hat{X}), \text{ and } e(\sigma_{m_i}^{(3)}, g) = e(\sigma_{m_i}^{(2)}, \hat{Y}\hat{Z}^{m_i}).$$

Next, $\mathcal{S}$ recomputes $\sigma_{m_i}^{\prime(3)} = \sigma_{m_i}^{(3)} \cdot (\sigma_{m_i}^{(2)})^{\sum_{j=1}^{n} \nu_j m_{i,j}}$ and returns valid signature $\sigma_{\vec{m}_i} = (\sigma_{m_i}^{(1)}, \sigma_{m_i}^{(2)}, \sigma_{m_i}^{\prime(3)})$ on $(pk, \vec{m}_i)$. It is easy to verify by using the verification algorithm of the CL+ signature scheme.

In the forgery phase, suppose $\mathcal{A}$ returns a valid forged signature $\sigma_{\vec{m}^*}$ on $(pk, \vec{m}^* = (m_1^*, \ldots, m_n^*))$. $\mathcal{S}$ computes $m^* = \sum_{i=1}^{n} \mu_i m_i^*$. $\mathcal{S}$ aborts if m^* appears in the query phase, i.e., for some $i \in [q_s]$ that $m^* = m_i$. Otherwise, $\mathcal{S}$ sets $\sigma_{m^*} = (\sigma_{m^*}^{(1)}, \sigma_{m^*}^{(2)}, \sigma_{m^*}^{(3)})$ as follows

$$\sigma_{m^*}^{(1)} = \sigma_{\vec{m}^*}^{(1)},$$
$$\sigma_{m^*}^{(2)} = \sigma_{\vec{m}^*}^{(2)},$$
$$\sigma_{m^*}^{(3)} = \sigma_{\vec{m}^*}^{(3)} \cdot (\sigma_{\vec{m}^*}^{(2)})^{-\sum_{i=1}^{n} \nu_i m_i^*}.$$

Therefore, $\mathcal{S}$ successfully forges a valid signature σ_{m^*} on (pk^*, m^*), which breaks the EUF-CMA security of simpleCL+ signature scheme.

The reduction $\mathcal{S}$ is indistinguishable from CL+ signature scheme, and public key elements $Z_1, \ldots, Z_n$ are random group elements in $\mathbb{G}$ from the point of view of the adversary since every randomness $\mu_1, \nu_1, \ldots, \mu_n, \nu_n \in \mathbb{Z}_p^*$ is information-theoretically hidden from the view of the adversary.

3.3. Efficiency and security

In this section, we discuss the efficiency and security comparison among the CL-like signatures under the EUF-CMA security model. The CL signatures [1] were first proven under the LRSW assumption [22]. While the LRSW assumption is an interactive assumption, which may not be desirable, its security was revisited and proven under the well-known, DLog assumption in the algebraic group model (AGM) [19]. Unfortunately, CL signatures suffer from the linear size growth issue, i.e. signature size is $(1+2n) \times \mathbb{G}$ where n is the number of messages, and the security reduction is non-tight even in AGM. On the other hand, the modified CL signature scheme [23] is another variant that is proven under the non-interactive, variant of q-SDH assumption [24], namely q-MSDH assumption in Definition 4. However, the modified CL signature is weakly randomizable because the additional element $\mathbb{Z}_p$ cannot be randomized, and the signature size is larger than CL signatures with the linear growth issue, i.e. signature size is $(3+2n) \times \mathbb{G} + 1 \times \mathbb{Z}_p$.

Based on the above discussion, we claim that our proposed CL+ signature scheme achieves constant size signature, i.e. signature size is $3\times\mathbb{G}$ regardless of the number of messages. The proposed CL+ signature scheme is the first variant that has been proven under DLog assumption in AGM with a tight reduction. The comparison is summarized in Table 2.

4. Anonymous credential systems

Based on the work in [1,20], the anonymous credential system can be obtained by adopting the following two protocols: (i) the signing committed messages protocol, in which a user can obtain signatures without revealing the messages; and (ii) the proving knowledge of the signatures protocol, in which a user can show the validity of signatures without revealing the plain message and signature pairs.

In the following section, we show that the proposed signature scheme in Section 3 can be adopted to construct the two aforementioned protocols. Note that we apply the (Generalized) Pedersen's commitment scheme [51] as defined in Section 2.5 and the sigma protocol as defined in Section 2.6 as the underlying building blocks. The underlying (Generalized) Pedersen's commitment scheme, i.e. $\mathtt{GP} = \{\mathtt{Commit}, \mathtt{Open}\}$, ensures that messages are information-theoretically hidden. And the underlying sigma protocol, i.e. Σ.PoK, allows one to prove the knowledge of (Generalized) Pedersen's commitment.

4.1. Signing committed messages protocol

This protocol is useful in the context of anonymous credential systems. It allows users to privately obtain signatures (credentials) without revealing messages (attributes). Without loss of generality, we refer to credentials as signatures and attributes as messages.

- `KeyGen`: A key generation algorithm that run by the signer. It randomly selects $x, y, z_1, \ldots, z_n \in \mathbb{Z}_p^*$, computes $X = g^x, Y = g^y, Z_1 = g^{z_1}, \ldots, Z_n = g^{z_n}$, and returns a public and secret key pair (pk, sk), such that

  $$pk = (X, Y, Z_1, \ldots, Z_n), \quad sk = (x, y, z_1, \ldots, z_n).$$

  The signer obtains (pk, sk).
- `Protocol`: A user who wishes to obtain a signature on a vector of message $\vec{m} = (m_1, \ldots, m_n) \in \mathbb{Z}_p^n$ initiates the protocol.

  (U1): The user runs GS.Commit on input messages $\vec{m} = (m_1, \ldots, m_n)$ as follows. It selects a randomness $t \in \mathbb{Z}_p^*$ and commits $\vec{m}$, such that the commitment is defined as $C = g^t \cdot \prod_{i=1}^{n} Z_i^{m_i}$. The user forwards C to the signer.

  (PoK): The user wishes to prove the knowledge of $(t, m_1, \ldots, m_n)$. The user and signer run Σ.PoK $\rightarrow \pi \in \{0,1\}$ to prove the knowledge of commitment C. If the signer is convinced, i.e., $\pi = 1$, the signer proceeds the protocol.

  (S2): The signer signs C by selecting a randomness $r \in \mathbb{Z}_p^*$ and computing $D = (D_1, D_2, D_3)$ as follows

  $$D_1 = g^r, \quad D_2 = X^r, \quad D_3 = (C)^{xr} \cdot (D_2)^y.$$

  The signer returns D to the user.

  (U3): The user aborts if the following equation does not hold, such that

  $$e(D_2, g) = e(D_1, X).$$

  Otherwise, the user extracts signature $\sigma_{\vec{m}} = (\sigma_{\vec{m}}^{(1)}, \sigma_{\vec{m}}^{(2)}, \sigma_{\vec{m}}^{(3)})$ by setting

  $$\sigma_{\vec{m}}^{(1)} = D_1, \quad \sigma_{\vec{m}}^{(2)} = D_2,$$
  $$\sigma_{\vec{m}}^{(3)} = \frac{D_3}{(D_2)^t} = (Y \cdot \prod_{i=1}^{n} Z_i^{m_i})^{xr}.$$

  It is easy to see $\sigma_{\vec{m}}$ is a valid signature on $\vec{m} = (m_1, \ldots, m_i)$ as the following verification holds, such that

  $$e(\sigma_{\vec{m}}^{(3)}, g) = e(\sigma_{\vec{m}}^{(2)}, Y \cdot \prod_{i=1}^{n} Z_i^{m_i}).$$

Signature is correct and the corresponding messages remain hidden. Given $D = (D_1, D_2, D_3)$, we see

$$D_3 = (g^t \cdot \prod_{i=1}^{n} g^{m_i z_i})^{xr} \cdot g^{xyr}$$
$$= g^{xrt} g^{xr(y+\sum_{i=1}^{n} m_i z)}.$$

The user can compute a valid signature element $\sigma_{\vec{m}}^{(3)}$ if secret $t \in \mathbb{Z}_p^*$ is known, such that

$$\begin{aligned}\sigma_{\vec{m}}^{(3)} &= \frac{D_3}{(D_2)^t} \\ &= \frac{g^{xrt} g^{xr(y+\sum_{i=1}^n m_i z)}}{g^{xrt}} \\ &= g^{xr(y+\sum_{i=1}^n m_i z)} \\ &= (Y \prod_{i=1}^n Z_i^{m_i})^{xr}.\end{aligned}$$

Therefore, the user obtains a valid signature $\sigma_{\vec{m}} = (\sigma_{\vec{m}}^{(1)}, \sigma_{\vec{m}}^{(2)}, \sigma_{\vec{m}}^{(3)})$ on $\vec{m}$ without revealing $\vec{m}$ to the signer as $\vec{m}$ is perfectly hidden by randomness g^t in (Generalized) Pedersen's commitment $C = g^t \cdot \prod_{i=1}^n Z_i^{m_i}$.

Security of the signing committed messages protocol. The security of the protocol relies on the EUF-CMA of the proposed CL+ signature scheme.

4.2. Proving knowledge of signatures protocol

Again, in the context of the anonymous credential system, once users obtain the signatures (credentials), it is desirable that users can prove the possession of the signatures without revealing the credential and attribute pairs. Throughout this protocol, we also generalize them as signature and message pairs. This proving knowledge of signatures protocol can be obtained based on the framework described in [1,20]. For concreteness, we detail the protocol as follows.

Let $(X, Y, Z_1, \dots, Z_n)$ be the signer's public key, and $\sigma_{\vec{m}} = (\sigma_{\vec{m}}^{(1)}, \sigma_{\vec{m}}^{(2)}, \sigma_{\vec{m}}^{(3)})$ be an valid signature on $\vec{m} = (m_1, \dots, m_n)$ based on the proposed CL+ signature scheme, where for some randomness $r \in \mathbb{Z}_p^*$,

$$\begin{aligned}\sigma_{\vec{m}}^{(1)} &= R = g^r, \\ \sigma_{\vec{m}}^{(2)} &= g^{xr}, \\ \sigma_{\vec{m}}^{(3)} &= g^{xr(y+\sum_{i=1}^n z_i m_i)}.\end{aligned}$$

To prove knowledge of $\sigma_{\vec{m}}$, the user (prover) initiates the following protocol.

1. (P1): The prover randomly selects $r', t \in \mathbb{Z}_p^*$ and computes a blinded signature $\sigma'_{\vec{m}} = (\sigma'^{(1)}_{\vec{m}}, \sigma'^{(2)}_{\vec{m}}, \sigma'^{(3)}_{\vec{m}})$, such that

   $$\begin{aligned}\sigma'^{(1)}_{\vec{m}} &= (\sigma_{\vec{m}}^{(1)})^{r'}, \\ \sigma'^{(2)}_{\vec{m}} &= (\sigma_{\vec{m}}^{(2)})^{r'}, \\ \sigma'^{(3)}_{\vec{m}} &= (\sigma_{\vec{m}}^{(3)})^{r' \cdot t^{-1}}.\end{aligned}$$

   The prover sends $\sigma'_{\vec{m}}$ to the verifier.
2. (V2): The verifier proceeds if the following verification algorithm holds, such that

   $$e(\sigma'^{(2)}_{\vec{m}}, g) = e(\sigma'^{(1)}_{\vec{m}}, X).$$

3. (PoK): The prover wishes to show the correctness of the signature without revealing $\vec{m}$. Let $V = e(g, g) \in \mathbb{G}_T$. The prover and verifier both compute $V_{xy}, V_{xz_1}, \dots, V_{xz_n}, V_s \in \mathbb{G}_T$ as follows, such that

   $$\begin{aligned}&V_{xy} = e(\sigma'^{(2)}_{\vec{m}}, Y), \quad V_s = e(\sigma'^{(3)}_{\vec{m}}, g), \\ &V_{xz_1} = e(\sigma'^{(2)}_{\vec{m}}, Z_1), \dots, V_{xz_n} = e(\sigma'^{(2)}_{\vec{m}}, Z_n).\end{aligned}$$

   The prover runs GS.Commit, which sets commitment be $C = (V_s)^t \cdot \prod_{i=1}^n (V_{xz_i})^{m_i}$ using its knowledge of $t, \vec{m} = (m_1, \dots, m_n)$. The verifier obtains C and carries out a (slightly modified) sigma protocol Σ.PoK $\rightarrow \pi$ with the prover. To prove the blinded signature $\sigma'_{\vec{m}}$ over $\mathbb{G}_T$, we detail how Σ.PoK works in $\mathbb{G}_T$ as follows.

   (a) The prover randomly selects $t_0, t_1, \dots, t_n$ and forwards $M = (V_s)^{t_0} \cdot \prod_{i=1}^n (V_{xz_i})^{t_i}$ to the verifier.
   (b) The verifier replies a random $c \in \mathbb{Z}_p^*$ back to the prover.
   (c) The prover sets $s_0 = t_0 + c \cdot t$ and $\forall i \in [1, \dots, n] : s_i = t_i + c \cdot m_i$, and returns $(s_0, s_1, \dots, s_n)$ to the verifier.
   (d) The verifier returns $\pi = 1$ if the proof is convincing, such that

   $$(V_s)^{s_0} \prod_{i=1}^n (V_{xz_i})^{s_i} = C^c \cdot M.$$

Suppose the proof is convincing. We show that it also implies the signature $\sigma'_{\vec{m}}$ is valid on randomness $\hat{r} = r \cdot r' \in \mathbb{Z}_p^*$ due to $(V_s)^t = V_{xy} \cdot \prod_{i=1}^n (V_{xz_i})^{m_i}$, which indicates the final verification algorithm in CL+ signature scheme, i.e.,

$$\begin{aligned}(V_s)^t &= (e(g, g)^{x\hat{r}t^{-1}(y+\sum_{i=1}^n z_i m_i)})^t \\ &= e(g, g)^{x\hat{r}y} \cdot e(g, g)^{x\hat{r}\sum_{i=1}^n z_i m_i} \\ &= V_{xy} \cdot \prod_{i=1}^n (V_{xz_i})^{m_i},\end{aligned}$$

such that

$$e(\sigma'^{(3)}_{m_i}, g)^t = e(\sigma'^{(2)}_{m_i}, Y) \cdot \prod_{i=1}^n e(\sigma'^{(2)}_{m_i}, Z)^{m_i}.$$

Completeness. It is easy to verify $(V_s)^{s_0} \prod_{i=1}^n (V_{xz_i})^{s_i} = C^c \cdot M$ is complete based on the fact that $\sigma'_{\vec{m}}$ is a valid signature on $(t, m_1, \dots, m_n)$ under $(X, Y, Z_1, \dots, Z_n)$, such that

$$\begin{aligned}(V_s)^{s_0} \prod_{i=1}^n (V_{xz_i})^{s_i} &= (V_s)^{s_0+c\cdot t} \prod_{i=1}^n (V_{xz_i})^{s_i+c\cdot m_i} \\ &= (V_s^t \prod_{i=1}^n V_{xz_i}^{m_i})^c \cdot (V_s^{s_0} \prod_{i=1}^n V_{xz_i}^{s_i}) \\ &= C^c \cdot M.\end{aligned}$$

Theorem 3. *The above protocol is a zero-knowledge proof of knowledge of a signature on messages $(m_1, \dots, m_n)$.*

Proof of Theorem 3. The completeness of the protocol has shown above. The zero-knowledge property is obtained by first showing that the verifier receives values, e.g. $(A, B, C) \in \mathbb{G}^3$ from the prover at (P1), are independent of the actual signature $\sigma'_{\vec{m}} = (\sigma'^{(1)}_{\vec{m}}, \sigma'^{(2)}_{\vec{m}}, \sigma'^{(3)}_{\vec{m}})$, such that (A, B) are random that satisfy $e(B, g) = e(A, X)$, and C is a random group element. Suppose there is a simulator $\mathcal{S}$ who chooses random $\gamma', \gamma \in \mathbb{Z}_p^*$ and sets $A = g^{\gamma'}, B = (X^{\gamma'}), C = g^{\gamma}$. We see $\sigma'_{\vec{m}} = (\sigma'^{(1)}_{\vec{m}}, \sigma'^{(2)}_{\vec{m}}, \sigma'^{(3)}_{\vec{m}}) = (A, B, C)$ are distributed correctly, and hence step (P1) is simulatable. Note that the verifier will not abort at step (V2) as the verification equation holds. At the last step, since the protocol adopts the sigma protocol, it follows that there exists a simulator $\mathcal{S}'$ that returns the proof π. Therefore, if $\mathcal{S}$ is constructed in this way, we obtain a zero-knowledge simulator for this protocol.

Lastly, we show that the protocol is proof of knowledge. Suppose there exists a prover that after interacting with the verifier, the verifier's acceptance is non-negligible. We can design a knowledge extractor algorithm that, given access to such a prover, the algorithm outputs a valid message and signature pair $(\vec{m} = (m_1, \dots, m_n), \sigma_{\vec{m}})$. Hence, with the prover, we can run such an extractor for the proof of knowledge protocol and obtain values $(\gamma', \vec{m})$ that satisfies equation $V_s^{\gamma'} = V_{xy} \prod_{i=1}^n (V_{xz_i})^{m_i}$ if $\pi = 1$, which is also the last verification equation of the CL+ signature scheme. Therefore, the extractor can return a valid signature $\sigma_{\vec{m}} = (A, B, C^{\gamma'})$ on $\vec{m}$.

5. Conclusion

In this work, we proposed a variant of CL-like signatures that possess randomizability without the linear size drawback, in which our signature size has a constant of three group elements. We proved its standard

EUF-CMA security without relying on the interactive assumption, but a standard DLog assumption in the AGM. The security reduction is tightly reduced to the DLog problem with a loss factor of 3, by enabling three indistinguishable simulations.

With the adoption of the AGM in the context of security analysis, we identified that one can achieve more efficient signature schemes with a tight security reduction under some well-studied assumptions and standard security models. Although our proposed scheme is proven in the idealized model, similar to the random oracle model and generic group model, we believe the scheme in AGM may still be useful in some restricted, controlled environments.

CRediT authorship contribution statement

Jia-Chng Loh: Conceptualization, Methodology, Validation, Formal analysis, Writing – original draft. **Fuchun Guo:** Methodology, Formal analysis, Validation, Supervision. **Willy Susilo:** Investigation, Resources, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgment

Willy Susilo is supported by the ARC Australian Laureate Fellowship FL230100033. Fuchun Guo is supported by the ARC Future Fellowship FT220100046.

References

[1] J. Camenisch, A. Lysyanskaya, Signature schemes and anonymous credentials from bilinear maps, in: Advances in Cryptology – CRYPTO 2004, Springer, 2004, pp. 56–72.

[2] D. Schröder, How to aggregate the CL signature scheme, in: Computer Security–ESORICS 2011: 16th European Symposium on Research in Computer Security, Leuven, Belgium, September 12-14, 2011. Proceedings 16, Springer, 2011, pp. 298–314.

[3] A. Bender, J. Katz, R. Morselli, Ring signatures: Stronger definitions, and constructions without random oracles, in: Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3, Springer, 2006, pp. 60–79.

[4] J. Camenisch, S. Hohenberger, M.Ø. Pedersen, Batch verification of short signatures, in: Advances in Cryptology-EUROCRYPT 2007: 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Barcelona, Spain, May 20-24, 2007. Proceedings 26, Springer, 2007, pp. 246–263.

[5] S. Canard, D. Pointcheval, O. Sanders, J. Traoré, Divisible e-cash made practical, IET Inf. Secur. 10 (6) (2016) 332–347.

[6] D. Bernhard, G. Fuchsbauer, E. Ghadafi, N.P. Smart, B. Warinschi, Anonymous attestation with user-controlled linkability, Int. J. Inf. Secur. 12 (2013) 219–249.

[7] P. Bichsel, J. Camenisch, G. Neven, N.P. Smart, B. Warinschi, Get shorty via group signatures without encryption, in: Security and Cryptography for Networks: 7th International Conference, SCN 2010, Amalfi, Italy, September 13-15, 2010. Proceedings 7, Springer, 2010, pp. 381–398.

[8] D. Chaum, E.v. Heyst, Group signatures, in: Workshop on the Theory and Application of of Cryptographic Techniques, Springer, 1991, pp. 257–265.

[9] J. Camenisch, A. Lysyanskaya, An efficient system for non-transferable anonymous credentials with optional anonymity revocation, in: Advances in Cryptology – EUROCRYPT 2001, Springer, 2001, pp. 93–118.

[10] M.H. Au, W. Susilo, Y. Mu, Constant-size dynamic k-TAA, in: Security and Cryptography for Networks: 5th International Conference, SCN 2006, Maiori, Italy, September 6-8, 2006. Proceedings 5, Springer, 2006, pp. 111–125.

[11] J. Camenisch, T. Groß, Efficient attributes for anonymous credentials, ACM Trans. Inf. Syst. Secur. 15 (1) (2012) 1–30.

[12] S. Canard, R. Lescuyer, Protecting privacy by sanitizing personal data: a new approach to anonymous credentials, in: Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security, 2013, pp. 381–392.

[13] F. Baldimtsi, A. Lysyanskaya, Anonymous credentials light, in: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, 2013, pp. 1087–1098.

[14] S. Goldwasser, S. Micali, R.L. Rivest, A digital signature scheme secure against adaptive chosen-message attacks, SIAM J. Comput. 17 (2) (1988) 281–308.

[15] V. Shoup, Lower bounds for discrete logarithms and related problems, in: Advances in Cryptology – EUROCRYPT 1997, Springer, 1997, pp. 256–266.

[16] U. Maurer, Abstract models of computation in cryptography, in: IMA International Conference on Cryptography and Coding, Springer, 2005, pp. 1–12.

[17] T. Jager, J. Schwenk, On the analysis of cryptographic assumptions in the generic ring model, J. Cryptol. 26 (2) (2013) 225–245.

[18] D. Aggarwal, U. Maurer, Breaking RSA generically is equivalent to factoring, in: Advances in Cryptology – EUROCRYPT 2009, Springer, 2009, pp. 36–53.

[19] G. Fuchsbauer, E. Kiltz, J. Loss, The algebraic group model and its applications, in: Advances in Cryptology – CRYPTO 2018, Springer, 2018, pp. 33–62.

[20] D. Pointcheval, O. Sanders, Short randomizable signatures, in: Cryptographers' Track at the RSA Conference, Springer, 2016, pp. 111–126.

[21] K.G. Paterson, J.C. Schuldt, Efficient identity-based signatures secure in the standard model, in: Australasian Conference on Information Security and Privacy, Springer, 2006, pp. 207–222.

[22] A. Lysyanskaya, R.L. Rivest, A. Sahai, S. Wolf, Pseudonym systems, in: International Workshop on Selected Areas in Cryptography, Springer, 1999, pp. 184–199.

[23] D. Pointcheval, O. Sanders, Reassessing security of randomizable signatures, in: Cryptographers' Track at the RSA Conference, Springer, 2018, pp. 319–338.

[24] D. Boneh, X. Boyen, Short signatures without random oracles, in: Advances in Cryptology - EUROCRYPT 2004, Springer, 2004, pp. 56–73.

[25] M. Bellare, P. Rogaway, Random oracles are practical: A paradigm for designing efficient protocols, in: Proceedings of the 1st ACM Conference on Computer and Communications Security, 1993, pp. 62–73.

[26] R. Canetti, O. Goldreich, S. Halevi, The random oracle methodology, revisited, J. ACM 51 (4) (2004) 557–594.

[27] S. Goldwasser, Y.T. Kalai, On the (in) security of the Fiat-Shamir paradigm, in: 44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings, IEEE, 2003, pp. 102–113.

[28] J. Nick, T. Ruffing, Y. Seurin, MuSig2: simple two-round Schnorr multi-signatures, in: Advances in Cryptology – CRYPTO 2021, Springer, 2021, pp. 189–221.

[29] M. Bellare, W. Dai, Chain reductions for multi-signatures and the HBMS scheme, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2021, pp. 650–678.

[30] H. Kılınç Alper, J. Burdges, Two-round trip schnorr multi-signatures via delinearized witnesses, in: Advances in Cryptology – CRYPTO 2021, Springer, 2021, pp. 157–188.

[31] E. Crites, C. Komlo, M. Maller, How to prove schnorr assuming schnorr: Security of multi-and threshold signatures, Cryptol. ePrint Arch. (2021).

[32] K. Lee, H. Kim, Two-round multi-signature from okamoto signature, Cryptol. ePrint Arch. (2022).

[33] G. Fuchsbauer, A. Plouviez, Y. Seurin, Blind schnorr signatures and signed ElGamal encryption in the algebraic group model, in: Advances in Cryptology–EUROCRYPT 2020, Vol. 12106, Nature Publishing Group, 2020, p. 63.

[34] J. Kastner, J. Loss, J. Xu, On pairing-free blind signature schemes in the algebraic group model, in: IACR International Conference on Public-Key Cryptography, Springer, 2022, pp. 468–497.

[35] S. Tessaro, C. Zhu, Short pairing-free blind signatures with exponential security, Cryptol. ePrint Arch. (2022).

[36] R. Bacho, J. Loss, On the adaptive security of the threshold BLS signature scheme, Cryptol. ePrint Arch. (2022).

[37] C.-P. Schnorr, Efficient identification and signatures for smart cards, in: Conference on the Theory and Application of Cryptology, Springer, 1989, pp. 239–252.

[38] D. Boneh, B. Lynn, H. Shacham, Short signatures from the Weil pairing, J. Cryptol. 17 (4) (2004) 297–319.

[39] J.-C. Loh, F. Guo, W. Susilo, G. Yang, A tightly secure ID-based signature scheme under DL assumption in AGM, in: Australasian Conference on Information Security and Privacy, Springer, 2023, pp. 199–219.

[40] T. Mizuide, A. Takayasu, T. Takagi, Tight reductions for Diffie–Hellman variants in the algebraic group model, in: Topics in Cryptology – CT-RSA 2019, Springer, 2019, pp. 169–188.

[41] B. Bauer, G. Fuchsbauer, J. Loss, A classification of computational assumptions in the algebraic group model, in: Advances in Cryptology – CRYPTO 2020, Springer, 2020, pp. 121–151.

[42] L. Rotem, G. Segev, Algebraic distinguishers: from discrete logarithms to decisional uber assumptions, in: Theory of Cryptography Conference, Springer, 2020, pp. 366–389.

[43] C. Ge, Z. Liu, W. Susilo, L. Fang, H. Wang, Attribute-based encryption with reliable outsourced decryption in cloud computing using smart contract, IEEE Trans. Dependable Secure Comput. (2023).

[44] C. Ge, W. Susilo, Z. Liu, J. Baek, X. Luo, L. Fang, Attribute-based proxy re-encryption with direct revocation mechanism for data sharing in clouds, in: Proceedings of the ACM Turing Award Celebration Conference-China 2023, 2023, pp. 164–165.

[45] J. Katz, J. Loss, J. Xu, On the security of time-lock puzzles and timed commitments, in: Theory of Cryptography Conference, Springer, 2020, pp. 390–413.

[46] R.L. Rivest, A. Shamir, D. Wagner, Time-Lock Puzzles and Timed-Release Crypto, Technical Report, MIT Laboratory for Computer Science, 1996.

[47] T. Agrikola, D. Hofheinz, J. Kastner, On instantiating the algebraic group model from falsifiable assumptions, in: Advances in Cryptology–EUROCRYPT 2020, Vol. 12106, Nature Publishing Group, 2020, p. 96.

[48] M. Bellare, P. Rogaway, The exact security of digital signatures-how to sign with RSA and rabin, in: Advances in Cryptology – EUROCRYPT 1996, Springer, 1996, pp. 399–416.

[49] D. Hofheinz, T. Jager, Tightly secure signatures and public-key encryption, Des. Codes Cryptogr. 80 (1) (2016) 29–61.

[50] J. Katz, N. Wang, Efficiency improvements for signature schemes with tight security reductions, in: Proceedings of the 10th ACM Conference on Computer and Communications Security, 2003, pp. 155–164.

[51] T.P. Pedersen, Non-interactive and information-theoretic secure verifiable secret sharing, in: Annual International Cryptology Conference, Springer, 1991, pp. 129–140.

[52] J. Kastner, J. Loss, O. Renawi, Concurrent security of anonymous credentials light, revisited, Cryptol. ePrint Arch. (2023).