Efficient Controlled Signature for a Large Network with Multi Security-level Setting

Pairat Thorncharoensri*, Willy Susilo, and Joonsang Baek
University of Wollongong, Wollongong, Australia
{pairat, wsusilo, baek}@uow.edu.au

Received: August 1, 2019; Accepted: September 2, 2019; Published: September 30, 2019

Abstract

We present an efficient multi-level controlled signature. This primitive allows a signer to specify a security level to limit the accessibility of the signature and the message. The primitive works as follows. Let the security levels of a group of users defined in the ascending order, where "1" stands for the lowest security level and "10" represents the highest protection level, respectively. A signer signs on a message by setting a security layer "3", which it is indicated that all users who were authorized with a level of security greater than "3" can verify this signature while cannot verify it. Many existing primitives, such as designated verifier signature, hierarchical identity-based signatures, policy-based signature, and attribute-based signature, are shared similarities with this primitive. However, our construction for this primitive is unique, concise and efficient compared to those existing primitives.

Keywords: Attribute-based, signature, designated verifier, anonymity.

1 Introduction

In any particular organization, a good management structure (in hierarchical sense) and communication between the management level of are essential to effectively evaluate the company strategies, plan the actions to be taken in order to reach the business goals and execute the given command from the upper management level among the workforce effectively. In some occasional, communication in the organization is not only needed the integrity of message to be secured, but the ambiguity is also needed to provide deniability for the signer.

The paper of Jakobsson *et al.* in [23] points out the need of the deniability and proposed the designated verifier signatures. A signature in this notion does not provide the only authentication of a message, but it also provides the deniability property that allows the signer to deny the signature. Since the verifier can also generate such a signature, it cannot claim that it was signed by the signer. However, only the designated verifier can be convinced that the signature was indeed signed by the signer.

Consider a scenario where Alice, who is a sale manager in a branch of a large organization, would like to report the performance of sales under her management to the upper management level. Alice would like to limit the information to only people in the upper management level, but not to anyone below her level. Hence, the signature of Alice should be designated to everyone whose level is higher than her. Intuitively, the notion of designated verifier signature capture the above issue, however, Alice will need to generate a designated verifier signature on a message for each person in the upper management level, which is not so efficient.

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), 10(3):1-20, Sep. 2019
DOI:*10.22667/JOWUA.2019.09.30.001*

*Corresponding author: Pairat Thorncharoensri, Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, NSW 2522, Australia, Tel.+61-0242214033

Let's consider another similar scenario in the airline industry. Alice, who works as a flight scheduling administrator in an airline company, would like to inform the change of some flight setting details to the cabin crew and cockpit officers. Due to the nature of jobs, there is a chain of command which can be ordered from the highest position as captains, first officers, second officers, cabin managers, senior cabin attendants, and cabin attendants. However, the position in this chain of command is fixed but persons in that position are always changing. Hence, this type of command's chain is not in the hierarchical formation but it is the multi-level formation. For example, the safety regulation for accessing the cockpit was changed and Alice has to inform all cockpit officers and cabin managers but not senior cabin attendants, and cabin attendants. Noted that this information is sensitive and confidential which cannot be leaked to outside. The information must be signed and designated to every single related personal.

Another scenario in a wireless sensor network or/and mesh network, the information transmits among the nodes to reach its destination. In normal circumstances, a controller needs to send a message to every single node that has the same functions. To ensure the integrity, authenticity and ambiguity, every message needs a designated verifier signature on it. Consider that nodes are assigned in a hierarchical structure such that the node in the same or above level can verify or/and forward the command message, or executes it simultaneously. In this scenario, a designated verifier signature with the multi-verifier setting will reduce the cost of the communication in this setting.

In this work, we use the notion of *multi-level controlled signatures* to solve the aforementioned problem efficiently. Our scheme benefits from its short signature size. This scheme is suited for an establishment where there are many security layers required and communication cost is matter. Moreover, the size of the signature in our scheme is contained only two elements which are one element in G_1 and another element in $\mathbb{Z}_p$.

1.1 Related Work

The multi-level controlled signatures was first proposed in [42] to capture the need for authenticating messages to a specified group of verifiers that satisfy the required security level. This notion works around the concept of attribute-based signature and designated verifier signature. Its environment setting is in the multi security-level instead of attributed-based policy.

Bagga and Molva in [2] proposed the notion of policy-based cryptography which includes policy-based encryption schemes and policy-based signature schemes. Since policy-based signature schemes are closely related to our work, its concept is simplified as follows. Policy-based signatures are similar to attribute-based signatures, where a signer can only sign a message if he/she satisfies the required policy (or attributes). On a signer's point of view, policy-based signatures ensure message integrity and authenticity properties. Any verifier can verify the authenticity and integrity of the message, however, nobody can forge the authenticity and integrity of the signatures that have been signed by a signer that satisfies a stated policy. Later on, Bellare and Fuchsbauer formalized policy-based signatures in [3].

Following Bagga and Molva's research, a variant called policy-controlled signature scheme was proposed by Thorncharoensri, Susilo, and Mu in [43]. The property of credential collision resistance was formalized and has been applied in that work. In contrast to policy-based signatures, policy-controlled signatures will guard the verifiers, and hence, only a verifier who processes some credentials satisfied the required policy can verify the authenticity and integrity of the message and non-repudiability of the signer. These properties also are also applicable to our work.

The notion of a hierarchical identity-based cryptosystem (HIBC) is other related works. Many researches are involving HIBC such as [16, 19, 6, 9, 45, 11, 29]. The hierarchical identity-based encryption (HIBE) is a united concept between a hierarchical system and identity-based encryption scheme (IBE) [40, 7] where an identity at the level k of the hierarchical system can issue a private key for its descendant identity, but it cannot decrypt a message on behalf of another identity except its descendants. Our work has a slightly similar aspect but has a difference in key generations and distribution. In other words, every level has only one single branch, but it can generate multiple private keys for its branch in the multi-level controlled system.

A natural conversion from a HIBE scheme, the hierarchical identity-based signature (HIBS) scheme [16, 12, 19] also inherits its properties where the ancestor identity of the hierarchical system can issue a private key for its descendant identity but cannot sign a message on behalf of other identities except its descendants. The purpose of a HIBE system is to reduce the bottleneck in a large network, where the PKG of an IBE system is applied, and to limit the scope of key escrow. Nevertheless, similar to policy-based signature schemes, HIBS only provides message integrity and authenticity for a signer but not non-repudiation and authorization for a verifier.

A closely related work, Attribute-Based Signatures scheme (ABS), was introduced in 2008 by Maji et al. [30]. Their schemes in the standard model were later presented in [31]. The objective of their study is to propose a primitive that allows a signer, who want to reveal no information about his/her identity, to sign a message using only the specified attributes that he/she is possessed of. Likewise, only a signer whose set of keys from the authority predicated satisfaction of his/her attributes can sign a message. Moreover, only the predicated attributes have authenticated the message but none of the information regarding the identity of a signer himself/herself.

Based on the results of Maji et al's works, many variant ABS schemes were conferred [39, 17, 18, 14, 33, 34, 32, 38] Two independent works on the attribute-based signature with threshold predicate were presented by Shahandashti and Safavi-Naini [39], and Li et al. [26]. A revocable ABS with threshold predicate was proposed by Escala et al. [14]. They also introduced an adaptive unforgeability property for ABS schemes. Later, a constant size of an attribute-based signature with threshold predicate was proposed by Herranz et al. [18]. Thereafter, Sakai et al. [38] presented an ABS scheme with arbitrary circuits that claims to be efficient than ABS schemes proposed by Maji et al. in [32, 31] when the number of gates is increased. Their construction is based on the combination of a witness indistinguishable and an extractable non-interactive proof system and an existentially unforgeable signature scheme.

Designated verifier signature(DVS) schemes were introduced by Jakobsson, Sako, and Impagliazzo in [23]. A signature of these schemes does not only provide authentication of a message, but it also provides the deniability property that allows the signer to deny this signature (since the verifier can also generate such a signature). Hence, only the designated verifier can verify the signature on a message. Later, the topics on designated verifier signatures have been widely studied [25, 24, 27, 28, 21, 41, 20], nonetheless, none of these works is a solution for the scenarios mentioned since a designated verifier signature scheme for the multi verifiers does not exist. In 2012, Fan et al. in [15] proposed attribute-based strong designated-verifier signature scheme (ABSDVS) with revokable and anonymity properties. This primitive seems to provide a solution for the aforementioned problems. Nevertheless, its unforgeability does not provide the required property, namely the authentication property. In unforgeability model of Fan et al.'s scheme stated that its ABSDVS scheme is similar to a strong designated verifier signatures scheme, which ensures that only the designated verifiers and the original signer can produce an ABSDVS signature designated to the designated verifiers with the specific attributes. In Figure 1, it shows that a verifier can produce a valid signature without any knowledge of the private key of signer. Moreover, according to Fan et al.'s scheme, the other verifiers could verify this signature but it does not guarantee that it signed by the signer or not due to the source-hidden property. This is because the source-hidden property is originated from the strong designated verifier signature [37, 41, 44, 22] and it can only be proven only when they are merely two parties namely a signer and a designated verifier. Therefore, in multiple verifiers case, it cannot be proven. For example, a signer produces a signature on a message with some conditions that only a CEO or a CFO can verify this signature. The CEO cannot be sure that this signature is signed by a signer or it was generated by a CFO. The above example indicated that

ABSDVS schemes cannot solve the aforementioned problems.

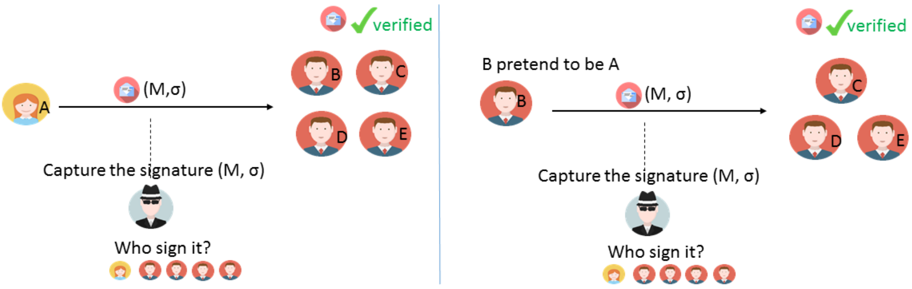

Figure 1: In ABSDVS scheme, both cases are true.

1.2 Our Contributions

To tackle the privacy issue over the information shared in a large organization, where a hierarchical structure implemented, an efficient (short) multi-level controlled signature (SMLCS) scheme has been proposed in this paper. The notion of MLCS scheme allows only receivers, who hold a credential for a certain security level specified by the signer, to verify the authenticity of the signed message. An MLCS scheme provides privacy, integrity, authenticity, and authority, and suitable for mass communication that require a privacy. The collision-resistance property in MLCS schemes provides a non-transferable right of verifying a signature, which is similar to the property required in a designated verifier signature (but not in a strong designated verifier signature).

Paper Organization
The organization of the paper is organized as follows. Some preliminaries that will be used throughout this paper is presented in the next section. The definition of MLCS and its security notions are described in the Section 3. A construction of the efficient MLCS scheme and its proofs are provided in Section 4. It is the most efficient compared to the MLCS scheme in [42] and it also occupies a very low communication overhead (the size of a signature are composed of only two elements). Finally, the comparison of the concrete schemes with other schemes and conclusion of the paper will be presented in the last two sections.

2 Preliminaries

2.1 Notation

The following notations will be used in the rest of this paper. A function $f : \mathbb{N} \rightarrow \mathbb{R}$ is *negligible* when, for all constant $c > 0$ and for all sufficiently large n, $f(n) < \frac{1}{n^c}$. $poly(.)$ is a deterministic polynomial function. Let $[n]$ represent a series of numbers(or indexes), e.g., if n is integer then $[n] = \{0,...,n\}$. Hence, for all polynomials $poly(k)$ and for all sufficiently large k, we say that q is polynomial-time in k if $q \leq poly(1^k)$. Denote by $l \overset{\$}{\leftarrow} L$ the operation of picking l at random from a (finite) set L. Let $H : \{0,1\}^* \rightarrow \mathbb{G}_1$ be a collision-resistant hash function. Let $h : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$ be a collision-resistant hash function.

2.2 Bilinear Pairing

We denote by $\mathbb{G}_1$ and $\mathbb{G}_2$ cyclic multiplicative groups where their generators are g_1 and g_2 respectively. We denote by p a prime and the order of both generators. Let $\mathbb{G}_T$ be another cyclic multiplicative group with the same order p. Let $\hat{e}$ be an efficient algorithm. We denote by $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ a bilinear mapping with the following properties:

1. *Bilinearity:* $\forall (g_1 \in \mathbb{G}_1; g_2 \in \mathbb{G}_2; a, b \in \mathbb{Z}_p) : \hat{e}(g_1^a, g_2^b) = \hat{e}(g_1, g_2)^{ab}$.

2. *Non-degeneracy:* $\exists\, g_1 \in \mathbb{G}_1 \exists\ g_2 \in \mathbb{G}_2 : \hat{e}(g_1, g_2) \neq 1$.

3. *Computability:* $\exists\, \hat{e} : \forall\, g_1 \in \mathbb{G}_1, \forall\, g_2 \in \mathbb{G}_2; \hat{e}(g_1, g_2) \in \mathbb{G}_T$

Note that there exists $\varphi(.)$ function which maps $\mathbb{G}_1$ to $\mathbb{G}_2$ or vice versa in one-time unit.

2.3 Complexity Assumptions

Definition 2.1 (Computational Diffie-Hellman (CDH) Problem)**.** Given a 3-tuple $(g, g^x,\ g^y \in \mathbb{G}_1)$ as input, output $g^{x \cdot y}$. An algorithm $\mathscr{A}$ has advantage ε' in solving the CDH problem if

$$\Pr[\mathscr{A}(g, g^x, g^y) = g^{x \cdot y}] \geq \varepsilon'$$

where the probability is over the random choice of $x, y \in \mathbb{Z}_q^*$ and the random bits consumed by $\mathscr{A}$.

Assumption 1. Computational Diffie-Hellman Assumption [13, 5] We say that the (t, ε')-CDH assumption holds if no PPT algorithm with time complexity $t(.)$ has an advantage at least ε' in solving the CDH problem.

Definition 2.2 (Decisional Bilinear Diffie-Hellman (DBDH) Problem)**.** Given a random 4-tuple $(g, g^x, g^y, g^z) \in \mathbb{G}_1$ and a random integer $Z \in \mathbb{G}_T$ as input, decide whether or not $Z = \hat{e}(g, g)^{x \cdot y \cdot z}$. An algorithm $\mathscr{A}$ is said to $(\mathtt{t}, \varepsilon')$ solves the DBDH problem in $\mathbb{G}_1, \mathbb{G}_T$, if $\mathscr{A}$ runs in time $\mathtt{t}$, and

$$\left| \Pr[\mathscr{A}(g, g^x, g^y, g^z, Z = \hat{e}(g, g)^{x \cdot y \cdot z}) = 1] - \Pr\left[\mathscr{A}\left(g, g^x, g^y, g^z, Z = \hat{e}(g, g)^d\right) = 1\right] \right| \geq \varepsilon',$$

where the probability is taken over the random choices of $x, y, z, d \in \mathbb{Z}_p,\ g \in \mathbb{G}_1$, and the random bits consumed by $\mathscr{A}$.

Assumption 2. Decisional Bilinear Diffie-Hellman Assumption We say that the (t, ε')-DBDH assumption in $\mathbb{G}_1, \mathbb{G}_T$ holds if there is no PPT algorithm that (t, ε') solves the DBDH problem.

3 Multi-level Controlled Signature Schemes (MLCS)

There are three main players in multi-level controlled signature (MLCS) schemes. A signer S generates a signature that can be verified *only* by a verifier V who holds a credential satisfying the multi-level security policy. The last player is a trusted authority TA who issues credentials associated with a security level in a multi-level security system. The verifier V uses these credentials to verify the signature. Let $\mathscr{LV}$ denote a security level in the multi-level security policy. Let $\mathscr{ML}$ be a multi-level security policy contained a level of security clearance of the verifier. For example, $\mathscr{ML}$ = "$\mathscr{LV} > n$" where n is the number indicating the security level. Noted that $\mathscr{ML}$ can be applied to another type of index or symbol to indicate the security level. Without losing generality, we assume that the order of the security levels increases, for example, a higher number means a higher security level[1].

[1] We note that for a decreasing order of security levels, our scheme can be slightly modified.

A multi-level controlled signature scheme Σ is a 6-tuple (*Setup*, *TKeyGen*, *SKeyGen*, *CreGen*, *Sign*, *Verify*), which is described as follows.

System Parameter Generation (*Setup*):

Given a security parameter ℓ as input, a probabilistic algorithm *Setup* outputs the system parameter param. That is,

$$Setup(1^{\ell}) \rightarrow \mathsf{param}.$$

***TA* Key Generator (*TKeyGen*):**

Given param as input, a probabilistic algorithm *TKeyGen* outputs the private key (sk_{TA}) and the public parameter (pk_{TA}) of a trusted authority. That is,

$$TKeyGen(\mathsf{param}) \rightarrow (pk_{TA}, sk_{TA}).$$

Signer Key Generator (*SKeyGen*):

Given param and pk_{TA} as input, a probabilistic algorithm *SKeyGen* outputs the private key (sk_S) and the public parameter (pk_S) of a signer. That is,

$$SKeyGen(\mathsf{param}, pk_{TA}) \rightarrow (pk_S, sk_S).$$

Verifier Credential Generator (*CreGen*):

Given param, sk_{TA} and an assertion $\mathscr{LV}$ indicated a security level of a verifier as input, a probabilistic algorithm *CreGen* outputs a verifier's credential . That is,

$$CreGen(\mathsf{param}, sk_{TA}, \mathscr{LV}) \rightarrow .$$

Multi-level Controlled Signature Signing (*Sign*):

Given param, pk_{TA}, sk_S, pk_S, a message M and the multi-level security policy $\mathscr{ML}$ as input, a probabilistic algorithm *Sign* outputs a signer's signature δ. That is,

$$Sign(\mathsf{param}, M, sk_S, pk_S, pk_{TA}, \mathscr{ML}) \rightarrow \delta.$$

Multi-level Controlled Signature Verification (*Verify*):

Given param, pk_{TA}, pk_S, $\mathscr{ML}$, , M and δ as input,a deterministic algorithm *Verify* outputs a verification decision $\mathsf{d} \in \{\mathtt{accept}, \mathtt{reject}\}$. That is,

$$Verify(\mathsf{param}, M, \delta, pk_{TA}, pk_S, \mathscr{ML},) \rightarrow \mathsf{d}.$$

3.1 Unforgeability Property

In MLCS, the unforgeability property means an attacker accessing the credential oracle cannot generate a multi-level controlled signature δ^* on a new message M^*. This model provides an assurance that, with access to the signing oracle $\mathscr{SSO}$, the verifying oracle $\mathscr{VCO}$, pk_S and pk_{TA}, no one should be able to produce a multi-level controlled signature on a new message M^* even if it arbitrarily chooses a multi-level security policy $\mathscr{ML}$, a message M and the entire credentials as input. The model was named a security against existential unforgeability under adaptive chosen message and credentials exposure attack $(EUF-CMCEA)$.

Before describing the formal definition of this model, some definitions are first defined. $\mathscr{A}_U$ is defined as the adaptively chosen message and credentials exposure adversary that attack the unforgeability of an MLCS scheme. Let $\mathscr{F}$ be a simulator. Noted that is defined as the credentials for the entire security level, for example, if the system has 12 security levels, then $= (_{1}, \ldots, _{12}, _{1}, \ldots, _{12})$. To precisely describe the ability of adversaries breaking the unforgeability of an MLCS scheme, $\mathscr{SSO}$ and $\mathscr{VCO}$ oracles are illustrated as below.

$\mathscr{SSO}(.,.)$ **:** At most ϱ_S times, when a query for a signature δ on its choice of a message M was issued, $\mathscr{SSO}$ runs the *Sign* algorithm to generate a signature δ on a message M corresponding with pk_{TA}, pk_S and $\mathscr{ML}$ and then returns δ.

$\mathscr{VCO}(.)$**:** At most ϱ_C times, when a query for the credential $_i$ corresponding to the arbitrarily chosen security level $\mathscr{LV}$ was issued, $\mathscr{VCO}$ responses with the corresponding credentials .

```
$\mathrm{Expt}^{EUF-CMCEA}_{\mathscr{A}_U,\Sigma}(\mathsf{k})$:
    $\mathsf{param} \xleftarrow{\$} \mathtt{Setup}(1^{\mathsf{k}})$ ; $(pk_{TA}, sk_{TA}) \xleftarrow{\$} TKeyGen(\mathsf{param})$ ; $(pk_S, sk_S) \xleftarrow{\$} SKeyGen(\mathsf{param}, pk_{TA})$
    $(\mathsf{st}, M, \mathscr{ML}) \xleftarrow{\$} \mathscr{A}_U{}^{\mathscr{SSO}(.,.),\mathscr{VCO}(.)}(\mathsf{choose}, pk_{TA}, pk_S, \mathsf{param})$;
    $\delta \xleftarrow{\$} \mathscr{A}_U{}^{\mathscr{VCO}(.)}(\mathsf{forge}, \mathsf{st}, M, \mathscr{ML})$
    If $sso(M, \mathscr{ML})$ has never been exectued;$Verify(\mathsf{param}, M, \delta, pk_{TA}, pk_S, \mathscr{ML},) = 1$
        Return 1
    EndIf
    Return 0
```

Figure 2: Experiment used to define unforgibility

Let us now proceed the formalization of unforgeability. To any multi-level controlled signature, an adversary $\mathscr{A}_U$ associated with the experiment was given in Figure 2. $\mathscr{A}_U$ has two functions namely a choose stage and a forge stage. With an adaptive strategy, $\mathscr{A}_U$, in the choose stage, arbitrarily chooses a message and makes queries to the signing oracle $\mathscr{SSO}(.,.)$ and the credential oracle $\mathscr{VCO}(.)$. The query processes are allowed repeatedly according to $\mathscr{A}_U$'s strategies. At the end of the stage, $\mathscr{A}_U$ outputs a message M and a multi-level security policy $\mathscr{ML}$ along with some state information(st) to be used in the forge state. In the second stage, $\mathscr{A}_U$ takes $M, \mathscr{ML}, \mathsf{st}$ as an input and outputs a valid multi-level controlled signature δ. $\mathscr{A}_U$ wins the above experiment if

1. $\mathscr{A}_U$ outputs a forged signature δ on a new message M with respect to pk_S and $\mathscr{ML}$.

2. $\mathtt{accept} \leftarrow Verify(M, \delta, pk_S, \mathscr{ML},)$.

3. $\mathscr{A}_U$ never makes a request for a multi-level controlled signature with $M, pk_S, \mathscr{ML}$ to the $\mathscr{SSO}$ oracle.

Let us denote $\mathsf{ADV}_{EUF-CMCEA}(.)$ as the success probability of $\mathscr{A}_U$ winning the above experiment.

Definition 3.1. An MLCS scheme is ($\mathtt{t}$,ϱ_S,ϱ_C,ε)-secure existential unforgeable under a chosen message and credentials exposure attack if there is no PPT adversary $\mathscr{A}_U$ such that the success probability

$$\mathsf{ADV}_{EUF-CMCEA}(\mathsf{k}) = \Pr\left[\mathrm{Expt}^{EUF-CMCEA}_{\mathscr{A}_U,\Sigma}(\mathsf{k}) = 1\right] = \varepsilon$$

is non-negligible in k, , where $\mathscr{A}_U$ runs in time at most $\mathtt{t}$, and makes at most ϱ_S signing queries and ϱ_C credential queries.

3.2 Coalition-resistance Property

The definition coalition-resistance was first formalized by Thorncharoensri et al. in [43]. In general, the unforgeability property in an MLCS scheme implies security against the coalition-resistance's attacker trying to forge the signature. In the unforgeability game, the attacker possesses the credentials of the entire security level, therefore, the ability of this attacker implies the ability of coalition-resistance's attacker. Consequently, the MLCS's coalition-resistance property has redefined the aim to prevent a group of corrupted credential holders (malicious verifiers) verifying a multi-level controlled signature δ^* on a message M^* with a multi-level security policy $\mathscr{M}\mathscr{L}$, where these malicious verifiers do not have enough credentials to satisfy the security level indicated in $\mathscr{M}\mathscr{L}$. Intuitively, given two signatures that one of the signatures is a valid signature for a certain multi-level security level while another is not, a malicious verifier should not be able to identify which one of the signatures is valid.

Before describing the formal definition of this model, some definitions are first defined. $\mathscr{A}_C$ is defined as the adaptive chosen message and chosen multi-level security policy distinguisher. *ECR* is denoted as the existential coalition-resistance property of an MLCS scheme. The signing oracle $\mathscr{S}\mathscr{S}\mathscr{O}$ and credential generator oracle $\mathscr{V}\mathscr{C}\mathscr{O}$ are used to describe the abilities of $\mathscr{A}_C$ breaking the coalition-resistance property. The $\mathscr{S}\mathscr{S}\mathscr{O}$ and $\mathscr{V}\mathscr{C}\mathscr{O}$ oracle have been described in 3.1 The formalization of the coalition-resistance is described as follows. To any multi-level controlled signature, an adversary $\mathscr{A}_C$ associated with the coalition-resistance experiment was given in Figure 3. $\mathscr{A}_C$ has two functions that are a challenge stage and a guess stage. With an adaptive strategy, $\mathscr{A}_C$, in the challenge stage, arbitrarily chooses a message and makes queries to the signing oracle $\mathscr{S}\mathscr{S}\mathscr{O}(.,.)$ and the credential oracle $\mathscr{V}\mathscr{C}\mathscr{O}(.)$. The query processes are allowed repeatedly according to $\mathscr{A}_C$'s strategies. At the end of the stage, $\mathscr{A}_C$ outputs a message M, a multi-level security policy $\mathscr{M}\mathscr{L}$ along with some state information(st) to be used in the guess state. The experiment randomly outputs a valid multi-level controlled signature δ or random strings, according to a bit b. In the second stage (Phase 2), $\mathscr{A}_C$ takes $M, \mathscr{M}\mathscr{L}, \mathsf{st}, \delta$ and outputs `accept` or `reject` (1 or 0). Note that if any of the conditions below is not satisfied then the experiment will be aborted.

```
$\mathrm{Expt}^{ECR-b}_{\mathscr{A}_C,\Sigma}(\mathsf{k})$:
  Phase 1:
  $\mathsf{param} \xleftarrow{\$} \mathtt{Setup}(1^{\mathsf{k}})$ ; $(pk_{TA}, sk_{TA}) \xleftarrow{\$} TKeyGen(\mathsf{param})$ ; $(pk_S, sk_S) \xleftarrow{\$} SKeyGen(\mathsf{param}, pk_{TA})$
  $(\mathsf{st}, M, \mathscr{M}\mathscr{L}) \xleftarrow{\$} \mathscr{A}_C^{\mathscr{S}\mathscr{S}\mathscr{O}(.,.),\mathscr{V}\mathscr{C}\mathscr{O}(.)}(\mathsf{challenge}, pk_{TA}, pk_S, \mathsf{param})$;
  If $b = 0$, then
    $\delta \xleftarrow{\$} \{0,1\}^*$
  Else
    $\delta \xleftarrow{\$} Sign(\mathsf{param}, M, sk_S, pk_S, pk_{TA}, \mathscr{M}\mathscr{L})$
  EndIF
  Phase 2:
  $d \xleftarrow{\$} \mathscr{A}_C^{\mathscr{V}\mathscr{C}\mathscr{O}(.)}(\mathsf{guess}, \mathsf{st}, M, \mathscr{M}\mathscr{L}, \delta)$
  If $vco(\mathscr{L}\mathscr{V} = min(\mathscr{M}\mathscr{L})), sso(M, \mathscr{L}\mathscr{V})$ have never been exectued.
    Return d
  EndIf
  Return $\perp$
```

Figure 3: Experiment used to define the coalition-resistance

1. $\mathscr{A}_C$ never issues a query for a multi-level controlled signature with $\mathscr{ML}$ and M as input to the $\mathscr{SSO}$ oracle.
2. With a restriction that $\mathscr{ML} =$ "$\mathscr{LV} \geq l$", $\mathscr{A}_C$ can only make a request for credentials that a security level $\mathscr{LV} < l$ to the $\mathscr{VCO}$ oracle.

Let $\mathsf{ADV}_{ECR}(.)$ be the success probability of $\mathscr{A}_C$ braking the coalition-resistance property of MLCS.

Definition 3.2. An MLCS scheme is $(\mathtt{t}, \varrho_S, \varrho_C,\ \varepsilon)$-secure existential coalition-resistant under a chosen message and chosen multi-level security policy attack if there is no PPT distinguisher $\mathscr{A}_C$ such that the success probability

$$\mathsf{ADV}_{ECR}(\mathsf{k}) = |\Pr\left[\mathrm{Expt}^{ECR-0}_{\mathscr{A}_C,\Sigma}(\mathsf{k})\right] - \Pr\left[\mathrm{Expt}^{ECR-1}_{\mathscr{A}_C,\Sigma}(\mathsf{k})\right]| = \varepsilon$$

is non-negligible in k, where $\mathscr{A}_C$ runs in time at most $\mathtt{t}$, and makes at most ϱ_S signing queries, and ϱ_C credential queries.

4 The Short Multi-level Controlled Signature Scheme (SMLCS)

The scheme is described as follows.

Setup **:** On input a security parameter ℓ, a trusted third party randomly chooses a prime $p \approx poly(1^\ell)$. Let $\mathbb{G}_1$, $\mathbb{G}_1$ and $\mathbb{G}_T$ denote three groups of prime order p. Let $\hat{e}$ be the bilinear mapping function, which maps $\mathbb{G}_1$ and $\mathbb{G}_2$ to $\mathbb{G}_T$. The above mapping function is defined as $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ Choose a random generator $g \in \mathbb{G}_1$, $o \in \mathbb{G}_2$ and a bilinear mapping function $\hat{e}$. Select a hash function $h(.)$ Let us denote by $\mathsf{param} = (p, \hat{e}, g, o, h)$ a system parameter. Then, *Setup* returns param.

TKeyGen **:** Let n be a number of security levels. On input a system parameter param, a trusted authority TA randomly generates a private key sk_{TA} and a public key pk_{TA} for each security level as follows: select random integers $\mu_0, ..., \mu_n, \gamma_0, ..., \gamma_{n+1}, a, b \in \mathbb{Z}_p$. Let $pk_{TA} = (U_1 = o^{\mu_1 \cdot a}, ..., U_n = o^{\mu_n \cdot a}, V_0 = g^{\gamma_0}, ..., V_n = g^{\gamma_n}, W_1 = g^{\sum_{i=1}^{n} \gamma_i \cdot \mu_i \cdot a + \gamma_0 \cdot \mu_0 \cdot a + b}, ..., W_n = g^{\sum_{i=n}^{n} \gamma_i \cdot \mu_i \cdot a + \gamma_0 \cdot \mu_0 \cdot a + b})$ denote a public key. Then, *TKeyGen* returns $sk_{TA} = (\mu_0, ..., \mu_n,\ \gamma_0, ..., \gamma_{n+1}, a, b)$ as a private key of the trusted authority and $pk_{TA} = (\mathbb{U}, \mathbb{V}, \mathbb{W})$ as a public key of the trusted authority where the vectors $\mathbb{U}$, $\mathbb{V}$ and $\mathbb{W}$ are $(U_1, ..., U_n)$, $(V_0, ..., V_n)$ and $(W_1, ..., W_n)$, respectively.

SKeyGen **:** On input a system parameter param and a public key of the trusted authority, *SKeyGen* randomly generates a private key sk_S and a public key pk_S as follows: First, choose a random integer $x \in \mathbb{Z}_p$. Let $\mathbb{X} = (X_0 = V_0^x, ..., X_n = V_n^x)$. Then, *SKeyGen* set $sk_S = x$ as a private key of the signer and $pk_S = \mathbb{X}$ as a public key of the signer. Finally, *SKeyGen* returns sk_S, pk_S.

CreGen **:** Let $\mathscr{LV}$ indicate a security level of a verifier, for example, $\mathscr{LV} =$"D" or "5". On input param pk_{TA}, sk_{TA} and a security level of a verifier $\mathscr{ML} = l$ that a verifier is satisfied to obtain, *CreGen* randomly generates a credential sk_{CR}, as follows: *CreGen* randomly selects $\nu_0 \in \mathbb{Z}_p^*$ and computes a credential at a security level of $\mathscr{ML} = l$, where $\nu_l = ((\gamma_0 \cdot \mu_0 + \gamma_l \cdot \mu_l) \cdot a + b - \nu_0 \cdot \gamma_0)/\gamma_l$; $C_0 = o^{\nu_0}$; $C_l = o^{\nu_l}$. *CreGen* returns $_V = (C_0, C_l)$ to the verifier as a credential for a security level assertion $\mathscr{LV} = l$.

Sign **:** Given param, pk_{TA}, sk_S, pk_S, $\mathscr{ML} =$ "$\mathscr{LV} \geq l$" and a message M, *Sign* computes a multi-level controlled signature σ on a message M as follows:

$$r \overset{\$}{\leftarrow} \mathbb{Z}_p,\ R = \hat{e}(W_l, o^r),\ \sigma_1 = h(R||pk_S||pk_{TA}||\mathscr{ML}||M),\ \sigma_2 = o^{\sigma_1 \cdot x + r},$$

The multi-level controlled signature on a message M is $\sigma = (\sigma_1, \sigma_2)$. *Sign* responses with σ.

Verify **:** Let a verifier possesses a credential for a security level assertion $\mathscr{LV} = k$, where $k \geq l$. Given pk_S, pk_{TA}, v, $\mathscr{ML} =$ "$\mathscr{LV} \geq l$", σ and a message M, *Verify* checks whether the following equations hold or not.

$$\sigma_1 \overset{?}{=} h(\hat{e}(W_l, \sigma_2) \cdot \hat{e}(X_0, C_0)^{-\sigma_1} \cdot \hat{e}(X_k, C_k)^{-\sigma_1} \prod_{i=l; i \neq k}^{n} \hat{e}(X_i, U_i)^{-\sigma_1} || pk_S || pk_{TA} || \mathscr{ML} || M).$$

If it does not hold, then *Verify* outputs `reject`. Otherwise, it outputs `accept`.

4.1 Security Analysis

4.1.1 Completeness

The signature verification is described as follows:

$$\begin{aligned}
\sigma_1 &\overset{?}{=} h(\hat{e}(W_l, \sigma_2) \cdot \hat{e}(X_0, C_0)^{-\sigma_1} \cdot \hat{e}(X_k, C_k)^{-\sigma_1} \\
&\quad \prod_{i=l; i \neq k}^{n} \hat{e}(X_i, U_i)^{-\sigma_1} || pk_S || pk_{TA} || \mathscr{ML} || M). \\
h(R || pk_S || pk_{TA} || \mathscr{ML} || M) &\overset{?}{=} h(\hat{e}(W_l, \sigma_2) \cdot \hat{e}(X_0, C_0)^{-\sigma_1} \cdot \hat{e}(X_k, C_k)^{-\sigma_1} \\
&\quad \prod_{i=l; i \neq k}^{n} \hat{e}(X_i, U_i)^{-\sigma_1} || pk_S || pk_{TA} || \mathscr{ML} || M).
\end{aligned}$$

From the above equation R is the point of interest for the completeness. Hence, the verification for R can be illustrated as follows:

$$\begin{aligned}
R = \hat{e}(W_l, o^r) &\overset{?}{=} \hat{e}(W_l, \sigma_2) \cdot \hat{e}(X_0, C_0)^{-\sigma_1} \cdot \hat{e}(X_k, C_k)^{-\sigma_1} \prod_{i=l; i \neq k}^{n} \hat{e}(X_i, U_i)^{-\sigma_1}. \\
\hat{e}(W_l, o^r) &\overset{?}{=} \hat{e}(W_l, o^{\sigma_1 \cdot x + r}) \cdot \hat{e}(X_0, C_0)^{-\sigma_1} \cdot \hat{e}(X_k, C_k)^{-\sigma_1} \prod_{i=l; i \neq k}^{n} \hat{e}(X_i, U_i)^{-\sigma_1}. \\
\hat{e}(W_l, o^{\sigma_1 \cdot x}) &\overset{?}{=} \hat{e}(X_0, C_0)^{\sigma_1} \cdot \hat{e}(X_k, C_k)^{\sigma_1} \prod_{i=l; i \neq k}^{n} \hat{e}(X_i, U_i)^{\sigma_1}. \\
\hat{e}(g^{\sum_{i=l}^{n} \gamma_i \cdot \mu_i \cdot a + \gamma_0 \cdot \mu_0 \cdot a + b}, o^{\sigma_1 \cdot x}) &\overset{?}{=} \hat{e}(g^{\gamma_0 \cdot x}, o^{\nu_0})^{\sigma_1} \cdot \hat{e}(g^{\gamma_k \cdot x}, o^{((\gamma_0 \cdot \mu_0 + \gamma_k \cdot \mu_k) \cdot a + b - \nu_0 \cdot \gamma_0) / \gamma_k})^{\sigma_1} \\
&\quad \prod_{i=l; i \neq k}^{n} \hat{e}(g^{\gamma_i \cdot x}, o^{\mu_i \cdot a})^{\sigma_1}. \\
\hat{e}(g, o)^{\sum_{i=l}^{n} \gamma_i \cdot \mu_i \cdot a + \gamma_0 \cdot \mu_0 \cdot a + b) \cdot \sigma_1 \cdot x} &\overset{?}{=} \hat{e}(g, o)^{\nu_0 \cdot \gamma_0 \cdot x \cdot \sigma_1} \cdot \hat{e}(g, o)^{((\gamma_0 \cdot \mu_0 + \gamma_k \cdot \mu_k) \cdot a + b - \nu_0 \cdot \gamma_0) \cdot \sigma_1 \cdot x} \cdot \\
&\quad \hat{e}(g, o)^{\sum_{i=l; i \neq k}^{n} \gamma_i \cdot \mu_i \cdot a \cdot \sigma_1}. \\
\hat{e}(g, o)^{\sum_{i=l}^{n} \gamma_i \cdot \mu_i \cdot a + \gamma_0 \cdot \mu_0 \cdot a + b) \cdot \sigma_1 \cdot x} &\overset{?}{=} \hat{e}(g, o)^{\sum_{i=l}^{n} \gamma_i \cdot \mu_i \cdot a + \gamma_0 \cdot \mu_0 \cdot a + b) \cdot \sigma_1 \cdot x}.
\end{aligned}$$

4.1.2 Unforgeability

Theorem 4.1. *The short multi-level controlled signature scheme is existentially unforgeable under an adaptive chosen message and credential exposure attack if the CDH assumption holds in the random oracle model.*

Proof. Assume that there exists a forger algorithm $\mathscr{A}_U$ running the existential unforgeability game defined in Section 3.1. Then we will show that, by using $\mathscr{A}_U$, an adversary $\mathscr{F}$ solves the CDH problem.

We now begin with the construction of oracles. First, on input g_1, g_1^x and g_1^y as an instance of the CDH problem, $\mathscr{F}$ runs *Setup* and sets $g = g_1, o = g_1^y$ and obtains $\mathsf{param} = (p, \hat{e}, g, o, h)$. $\mathscr{F}$ then runs *TKeyGen* to obtains *TA*'s public-private keys and sets $\mathbb{X}^b = (X_0 = g_1^{x \cdot \gamma_0}, ..., X_n = g_1^{x \cdot \gamma_n}$ as the signer public key pk_S. Then, $\mathscr{F}$ constructs oracles as follows:

$\mathscr{H}\mathscr{O}$ **oracle:** On input a string Γ, if it is a request for a hash value of $h(\Gamma)$, $\mathscr{H}\mathscr{O}$ check whether Γ in the queried list or not. If it exists in the list then return the corresponding value, otherwise, $\mathscr{H}\mathscr{O}$ randomly chooses $\iota \overset{\$}{\leftarrow} \mathbb{Z}_p$ then returns $h(\Gamma) = \iota$. Noted that $\mathscr{H}\mathscr{O}$ keeps (Γ, ι) in the list and this list can be accessed only by $\mathscr{F}$.

$\mathscr{V}\mathscr{C}\mathscr{O}$ **queries :** On input a secret key sk_{TA}, $\mathscr{V}\mathscr{C}\mathscr{O}$ runs *CreGen* to generate the credential *VCR* for the security level assertion $\mathscr{L}\mathscr{V} = l$ and then returns *VCR*.

$\mathscr{S}\mathscr{S}\mathscr{O}$ **queries :** On input $\mathscr{M}\mathscr{L} =$ "$\mathscr{L}\mathscr{V} \geq l$" and a message M, $\mathscr{S}\mathscr{S}\mathscr{O}$ computes a multi-level controlled signature as follows:

$$\begin{array}{rcl} r, \iota & \overset{\$}{\leftarrow} & \mathbb{Z}_p, \\ R & = & \hat{e}(W_l^x, o)^\iota \hat{e}(W_l, o^r) : \ W_l^x = (g_1^x)^{\sum_{i=l}^{n} \gamma_i \cdot \mu_i \cdot a + \gamma_0 \cdot \mu_0 \cdot a + b}, \\ \sigma_1 & = & h(R || pk_S || pk_{TA} || \mathscr{M}\mathscr{L} || M) = \iota, \ \sigma_2 = o^r, \\ \Gamma & = & R || pk_S || pk_{TA} || \mathscr{M}\mathscr{L} || M. \end{array}$$

Noted that $\mathscr{S}\mathscr{S}\mathscr{O}$ has an access to the list of (Γ, ι) via $\mathscr{F}$. There, $\mathscr{S}\mathscr{S}\mathscr{O}$ uses this advantage to update (Γ, ι) to the list in $\mathscr{H}\mathscr{O}$. $\mathscr{S}\mathscr{S}\mathscr{O}$ then responds with $\sigma = (\sigma_1, \sigma_2)$.

Now, we begin the game by giving an access to the above oracles to $\mathscr{A}_U$. Assume that $\mathscr{A}_U$ always makes a query for a string or a message to $\mathscr{H}\mathscr{O}$ oracle before it outputs a potential forgery, denoted by $M^*, \sigma^*, \mathscr{M}\mathscr{L}^*$. After executing an adaptive strategy with the above oracles, $\mathscr{A}_U$ outputs a forgery σ^* on a message M^* with respect to $\mathscr{M}\mathscr{L}^*$. $\mathscr{A}_U$ wins the game if a multi-level controlled signature σ^* on the message M^* with respect to $\mathscr{M}\mathscr{L}^*$ is valid and is not an output from the $\mathscr{S}\mathscr{S}\mathscr{O}$ queries.

We denote by ε the success probability $\mathsf{ADV}_{EUF-CMCEA}(.)$ that $\mathscr{A}_U$ wins the game. Let e be the base of the natural logarithm. As we mentioned early, a query for a hash of a string or message to $\mathscr{H}\mathscr{O}$ is always issued before $\mathscr{A}_U$ issues a query for a signature to $\mathscr{S}\mathscr{S}\mathscr{O}$ queries, hence, $q_H \geq q_S$. In order to solve CHD problem, we using the Forking technique in [36, 4]. $\mathscr{F}$ first obtain a signature σ^* on message M^* where $\sigma_1^* = h(\Gamma^*) = \iota^*$. Next, $\mathscr{F}$ resets $\mathscr{A}_U$ to the initial state and repeats again the above experiment with $\mathscr{A}_U$ with a different hash value $\sigma_1^{'} = h(\Gamma^*) = \iota^{'}$. Hence, $\mathscr{A}_U$ will output another signature $\sigma^{'}$ on message M^* where $h(\Gamma^*) = \iota^{'}$. From those signatures, $\mathscr{F}$ compute

$$\begin{aligned}
(\frac{\sigma_2^*}{\sigma_2^{'}})^{\frac{1}{\sigma_1^* - \sigma_1^{'}}} &= (\frac{o^{\sigma_1^* \cdot x+r}}{o^{\sigma_1^{'} \cdot x+r}})^{\frac{1}{\sigma_1^* - \sigma_1^{'}}} \\
&= (\frac{(g_1^y)^{\iota^* \cdot x+r}}{(g_1^y)^{\iota^{'} \cdot x+r}})^{\frac{1}{\iota^* - \iota^{'}}} \\
&= ((g_1^y)^{\iota^* \cdot x+r-\iota^{'} \cdot x-r})^{\frac{1}{\iota^* - \iota^{'}}} \\
&= (g_1^{y \cdot x(\iota^* - \iota^{'})})^{\frac{1}{\iota^* - \iota^{'}}} \\
&= g_1^{y \cdot x}
\end{aligned}$$

Let ε' the success probability $\mathsf{ADV}_{CDH}(.)$ that $\mathscr{F}$ solves the CDH problem. From the Forking Lemma in [36, 4],analyze the success probability that $\mathscr{F}$ uses $\mathscr{A}_U$ to solve CDH problem as follows:

$$\begin{aligned}
\varepsilon' \geq \text{frk} &\geq \text{acc}(\frac{\text{acc}}{q_H} - \frac{1}{2^l}) \\
\text{frk} &\geq \varepsilon(\frac{\varepsilon}{q_H} - \frac{1}{2^l}) \\
\text{frk} &\geq \frac{\varepsilon^2}{q_H} - \frac{\varepsilon}{2^l} \\
\text{frk} &> \frac{\varepsilon^2}{q_H} \\
\varepsilon' &> \frac{\varepsilon^2}{q_H} \\
\therefore \varepsilon &< \sqrt{q_H \varepsilon'}
\end{aligned}$$

Since $\mathscr{F}$ behaves naturally and does not need to abort the experiment in any event, $acc = \varepsilon$. Noted that $\frac{\varepsilon}{2^l}$ is negligible, hence, it is omitted. To summarize the probability, $\mathscr{A}_U$ wins the above game and outputs a signature σ^* on a message M^* with a probability less than $\sqrt{q_H \varepsilon'}$. The above success probability shows that our multi-level controlled signature scheme secures against existentially unforgeable under an adaptive chosen message and credential exposure attack if the success probability of solving CDH problem is negligible. □

4.1.3 Coalition-resistant

Theorem 4.2. *The short multi-level controlled signature scheme is existentially coalition-resistant against the adaptively chosen message and chosen multi-level security policy distinguisher $\mathscr{A}_C$ if the DBDH assumption holds in the random oracle model.*

Proof. Assume that an adversary $\mathscr{A}_C$ runs the existentially coalition-resistant game defined in Section 3.2 and successfully outputs a correct guess. We then will show that an adversary $\mathscr{F}$ can solve the DBDH problem by using $\mathscr{A}_C$ as a tool. On input $\mathbf{g}$, $\mathbf{g^x}$, $\mathbf{g^y}$, $\mathbf{g^z}$ and $\mathbf{Z}$ as an instance of the DBDH problem, $\mathscr{F}$ runs

$Setup$ and sets $g = \mathbf{g}, o = \mathbf{g}^{\mathbf{y}}$ and obtains $\mathsf{param} = (p, \hat{e}, g, o, h)$. $\mathscr{F}$ then sets $b = \mathbf{z}$ and runs $TKeyGen$ to obtains TA's public-private keys. $\mathscr{F}$ also sets $x = \mathbf{x}$ and runs $SKeyGen$ to obtains the signer public key pk_S. Assume that there exists an algorithm managing the list of each queries and such algorithms will be omitted. $\mathscr{F}$ constructs the oracles as follows:

$\mathscr{H}\mathscr{O}$ **oracle:** On input a string Γ, if it is a request for a hash value of $h(\Gamma)$, $\mathscr{H}\mathscr{O}$ check whether Γ in the queried list or not. If it exists in the list then return the corresponding value, otherwise, $\mathscr{H}\mathscr{O}$ randomly chooses $\iota \xleftarrow{\$} \mathbb{Z}_p$ then returns $h(\Gamma) = \iota$. Noted that $\mathscr{H}\mathscr{O}$ keeps (Γ, ι) in the list and it can be accessed only by $\mathscr{F}$.

$\mathscr{V}\mathscr{C}\mathscr{O}$ **queries :** $\mathscr{F}$ randomly chooses a integer $d \xleftarrow{\$} \mathbb{Z}^*_{n+1}$. On input $\mathscr{L}\mathscr{V} = l$, if $l \geq d$ then output $\perp$. Otherwise, $\mathscr{V}\mathscr{C}\mathscr{O}$ randomly chooses a integer $k_c \in \mathbb{Z}_p$ and compute $_V$ as follows:

$$\begin{aligned} k_1 &\xleftarrow{\$} \mathbb{Z}_p : k_1 = k_2 + k_c \\ C_0 &= o^{k_2}, \\ C_l &= o^{((\gamma_0 \cdot \mu_0 + \gamma_l \cdot \mu_l) \cdot a + k_1 - k_2 \cdot \gamma_0)/\gamma_l}, \end{aligned}$$

$\mathscr{V}\mathscr{C}\mathscr{O}$ then returns $_V = (C_0, C_l)$.

$\mathscr{S}\mathscr{S}\mathscr{O}$ **queries :** On input $\mathscr{M}\mathscr{L} =$ "$\mathscr{L}\mathscr{V} \geq l$" and a message M, if $l \geq d$ then output $\perp$. Otherwise, $\mathscr{S}\mathscr{S}\mathscr{O}$ computes a multi-level controlled signature as follows:

$$\begin{aligned} r, \iota &\xleftarrow{\$} \mathbb{Z}_p, \\ R &= \hat{e}(W_l, o^r) \cdot \hat{e}(X_0, o^{\mu_0 \cdot a + k_c \cdot \gamma_0^{-1}})^{\iota} \prod_{i=l}^{n} \hat{e}(X_i, U_i)^{\iota} : \\ \sigma_1 &= h(R||pk_S||pk_{TA}||\mathscr{M}\mathscr{L}||M) = \iota,\ \sigma_2 = o^r, \\ \Gamma &= R||pk_S||pk_{TA}||\mathscr{M}\mathscr{L}||M. \end{aligned}$$

Noted that $\mathscr{S}\mathscr{S}\mathscr{O}$ has an access to the list of (Γ, ι) via $\mathscr{F}$. There, $\mathscr{S}\mathscr{S}\mathscr{O}$ uses this advantage to update (Γ, ι) to the list in $\mathscr{H}\mathscr{O}$. $\mathscr{S}\mathscr{S}\mathscr{O}$ then responds with $\sigma = (\sigma_1, \sigma_2)$.

At the beginning of a game, $\mathscr{A}_C$ is given with An access to the above oracles. Next, we run an experiment between $\mathscr{A}_C$ and $\mathscr{F}$ as modeled in Section 3.2 as follows:

1. **Phase** 1 **:** With any adaptive strategy, $\mathscr{A}_C$ arbitrarily makes queries to $\mathscr{S}\mathscr{S}\mathscr{O}$, $\mathscr{V}\mathscr{C}\mathscr{O}$ oracles. The oracles response as we mentioned in the above.
2. **Challenge :** At the end of the first phase, $\mathscr{A}_C$ decides to challenge and then outputs M^* and $\mathscr{M}\mathscr{L}^*$. $\mathscr{F}$ aborts the game if

   1. On input $\mathscr{M}\mathscr{L}^*$ and M^*, $\mathscr{A}_C$ issued a request for a multi-level controlled signature to $\mathscr{S}\mathscr{S}\mathscr{O}$ queries.

   2. $\mathscr{A}_C$ has a credential that equal or higher than the security level assigned to the multi-level security policy $\mathscr{M}\mathscr{L}^*$.

   Otherwise, $\mathscr{F}$ computes a response as follows:

$$\begin{aligned} r, \iota^* &\xleftarrow{\$} \mathbb{Z}_p, \\ R &= Z^{\iota^*} \cdot \hat{e}(W_l, o^r) \cdot \hat{e}(X_0, o^{\mu_0 \cdot a})^{\iota^*} \prod_{i=l}^{n} \hat{e}(X_i, U_i)^{\iota^*} : \\ \sigma_1^* &= h(R||pk_S||pk_{TA}||\mathscr{M}\mathscr{L}^*||M^*) = \iota,\ \sigma_2^* = o^r, \\ \Gamma^* &= R||pk_S||pk_{TA}||\mathscr{M}\mathscr{L}^*||M^*. \end{aligned}$$

Noted that $\mathscr{F}$ has an access to the list of (Γ^*, ι^*). There, $\mathscr{F}$ uses this advantage to update (Γ^*, ι^*) to the list in $\mathscr{HO}$. $\mathscr{F}$ then responds with $\sigma^* = (\sigma_1^*, \sigma_2^*)$ to $\mathscr{A}_C$.

3. **Phase** 2 **:** In this phase, $\mathscr{A}_C$ can go back to *Phase* 1 or *Challenge* as many as it requests. However, $\mathscr{F}$ will abort the game if

   1. On input $\mathscr{ML}^*$ and M^*, $\mathscr{A}_C$ issued a request for a multi-level controlled signature to $\mathscr{SSO}$ queries.

   2. $\mathscr{A}_C$ has a credential that equal or higher than the security level assigned to the multi-level security policy $\mathscr{ML}^*$.

4. **Guessing :** On the valid challenge $M^*, \mathscr{ML}^*, \sigma^*$, $\mathscr{A}_C$ finally outputs a guess b'.

Let $\mathsf{ADV}_{ECR} = \varepsilon$ be an advantage that $\mathscr{A}_C$ wins the above game. Let q be a polynomial upper bound of queries that $\mathscr{A}_C$ issues to the $\mathscr{HO}$ oracle. Note that $q \geq q_H$ and $q << p$. Since only $\mathscr{F}$ and $\mathscr{SSO}$ access $\mathscr{HO}$ before it outputs a response, thus, we can conclude that $q_H \geq q_S$. Therefore, we can analyze the advantage that $\mathscr{A}_C$'s guess is correct and wins the above game as follows:

- E_1: *$\mathscr{F}$ does not abort during the issuing of queries to the $\mathscr{VCO}$.* Let q_{VC} be the highest security level that $\mathscr{A}_C$ issued to the *vco* oracle rather than a number of queries that make to the *vco* oracle. Since $\mathscr{A}_C$ can make just one query for the security level $\mathscr{LV} = n-1$, $\mathscr{A}_C$ can use this credential to verify signatures with the entire security level except for the security level n. Note that d is a random integer chosen at the beginning of the game and n is the upper bound of the security level. The fact is that $\mathscr{A}_C$ can make a request for credential up to the security level $q_{VC} = n-1$ to the $\mathscr{VCO}$ oracle and the value of d is in range of $\{1, ..., n\}$. However, if $q_{VC} \geq d$, then the $\mathscr{VCO}$ will always terminate the experiment. Otherwise, $q_{VC} < d$, then the $\mathscr{VCO}$ will not terminate the experiment. To pick up q_{VC} and d randomly, the probability that $\mathscr{A}_C$ chooses q_{VC} is $\frac{1}{n}$ and the probability that $\mathscr{F}$ chose d is $\frac{1}{n}$ Therefore, the probability of this event is $\frac{1}{n^2}$.

- E_2: *$\mathscr{F}$ does not abort after* Phase 1 *and* Phase 2. Since we have assumed that $\mathscr{A}_C$ follows the experiment and outputs a guess with a valid challenge $(M^*, \mathscr{ML}^*, \sigma^*)$, then the probability of this event is 1.

The advantage that $\mathscr{A}_C$ wins the above game and outputs a correct guess $b' = b$ is

$$\Pr[\mathsf{ADV}_{ECR}] \cdot \Pr[\mathsf{ADV}_{ECR} | E_1 | E_2] \geq \varepsilon \frac{1}{n^2}.$$

Let ε' be an advantage to solve the DBDH problem. From the above game, $\mathscr{F}$ outputs a guess for the DBDH problem with $\mathscr{A}_C$'s guess. Note that $\mathscr{A}_C$ can choose a challenge multi-level security policy $\mathscr{ML}^*$, which $\mathscr{A}_C$ does not have a credential for that security level or above.

Hence, there is an event that the $\mathscr{A}_C$'s guess in the game is not the right guess for the DBDH problem, where $\mathscr{ML}^* \neq$ "$\mathscr{LV} \geq d$". The probability of this event is $\frac{1}{n}$. To conclude, an advantage that $\mathscr{F}$ outputs a correct guess for the DBDH problem by using $\mathscr{A}_C$ is $\varepsilon' \geq \varepsilon \cdot \frac{1}{n^2} \cdot \frac{1}{n} = \varepsilon \cdot \frac{1}{n^3}$. Hence, the advantage that $\mathscr{A}_C$ breaks the existentially coalition-resistant of our MLCS scheme against the adaptively chosen message and chosen multi-level security policy attack is $\varepsilon \leq n^3 \varepsilon'$. Since $n << q_H << q$, the analysis of the above advantages shows that the success of breaking the existentially coalition-resistant of our MLCS scheme is non-negligible if the advantage of breaking the DBDH problem is non-negligible. □

5 Asymptotic Analysis and Experimental Results

We proposed the efficient multi-level controlled signature (MLCS) schemes to capture the need for authenticating messages to a specified group of verifiers that satisfy the required security level.

Version / Size&Comp.	First MLCS in [42]	Second MLCS in [42]	SMLCS
PK_{TA}	$(2n+2)\lvert\mathbb{G}_1\rvert$	$(n+3)\lvert\mathbb{G}_1\rvert$	$(3n)\lvert\mathbb{G}_1\rvert$
SK_{TA}	$(3n+2)\lvert p\rvert$	$(n+3)\lvert p\rvert$	$(2n+2)\lvert p\rvert$
PK_S	$3\lvert\mathbb{G}_1\rvert$	$(n+1)\lvert\mathbb{G}_1\rvert$	$n\lvert\mathbb{G}_1\rvert$
SK_S	$\lvert p\rvert$	$\lvert p\rvert$	$\lvert p\rvert$
VCR_V	$(2l)\lvert\mathbb{G}_1\rvert$	$2\lvert\mathbb{G}_1\rvert$	$2\lvert\mathbb{G}_1\rvert$
Signature	$6\lvert\mathbb{G}_1\rvert+2\lvert p\rvert$	$(5+n-l)\lvert\mathbb{G}_1\rvert+2\lvert p\rvert$	$\lvert\mathbb{G}_1\rvert+\lvert p\rvert$
Sign Comp.	$H+7E+M+P$	$H+(6+n-l)E+M+P$	$2E+P$
Verify Comp.	$H+E+2lM+10P$	$H+E+(2(n-l)+8)P$	$((n-l)+2)P$

Table 1: The comparison of three MLCS schemes.

Computation cost for the security of discrete log with 1024 bits	SMLCS	PS-IDS	BLS-SS
Signer Key Generation Computation Time (Avg)	273ms	45ms	22ms
Verifier Key Generation Computation Time (Avg)	46ms	N/A	N/A
Signature Generation Computation Time (Avg)	61ms	61ms	71ms
Verification Computation Time (Avg)	148ms	73ms	79ms
Computation cost for the security of discrete log with 2048 bits	SMLCS	PS-IDS	BLS-SS
Signer Key Generation Computation Time (Avg)	4310ms	766ms	372ms
Verifier Key Generation Computation Time (Avg)	740ms	N/A	N/A
Signature Generation Computation Time (Avg)	923ms	785ms	385ms
Verification Computation Time (Avg)	2229ms	1223ms	609ms

Table 2: The practical computation time comparison of SMLCS scheme, PS identity-based signatures [35] and BLS signature scheme [8].

The comparison between our scheme and the schemes in [42] is summarized in Table 1. Note that l is a security level in the multi-level security policy where $MP =$ "$\mathscr{LV} \geq l$". n is the number of security level. Let E denote a computation of exponential in G_1 or G_T. Let M be a computation of multiplication in G_1. Let P be a computation of bilinear pairing function $\hat{e}$. A computation for hash functions in G_1 denoted as H. Since computation for a hash function in $\mathbb{Z}_p$ is trivial, it is omitted.

We conducted an experiment to implement our schemes using the Java Pairing-Based Cryptography Library (JPBC) provided by [10]. The experiment was conducted on Intel Xeon CPU model X5650 with CPU clocked at 2.67 GHz with 2 cores and 4 threads configuration and 16 Gigabyte of memory. The operation system used in these experiments is Window 8.1. The experiment was executed with two different types of curves, which are Type A and Type A1. Type A curve is a curve that produces the fastest bilinear pairing computation and it achieved the security comparable to the 1024 bits of discrete logarithm (DLog) security. On another hand, Type A1 provides a higher security, which is 2048 bits of DLog security. The parameters of these two curves are provided in Table 3. The experiment was conducted 100 times for each scheme to find the average of times consumed in each computation process. A

Curve Type A	**value**
Base field size (bits)	512
k	2
DLog security (bits)	1024

q	8780710799663312522437781984754049815806883199414208211028
	65339926647563088022295707862517942266222142315585876958
	231745927771336731748132492512999822479 1

h	12016012264891146079388821366740534204802954401251311822919 6151310472072893597045311028448021839065377867 76
r	730750818665451621361119245571504901405976559617
exp1	107
exp2	159
sign0&sign1	1

Curve Type A1	**value**
Base field size (bits)	1024
k	2
DLog security (bits)	2048

q	485128758963037524997122772545896285164193521882945211981 8
	956751100907315811504536129483934709931589896004539852468 20
	073341649285315947991491005480364457601109131574206556903 61
	891290858441360807158247259460501343449199712532828063940 00
	868374004850098044198971373968965561057845838812693424263 05
	57397618776539259

l	1304

n (=r)	362036387285848899251584158616340511316562329763391949240 2
	206530672318892396645176216032787096963873056719805860050 89
	606971380063668617904097765283854072836648605652392952913 14
	844246909284597617282274074224254733917313218308080644731 34
	976398511082162719551471174603705642580481969263204047957 50
	42834043863089

Table 3: Curve Type A and Type A1 parameters used in PBC library[1].

messages use in the experiment has been randomly generated with the fixed size of 100 bytes. Without losing generality, we compare the computation cost of our SMLCS scheme with the provided example schemes in the JPBC library, which are Paterson-Schuldt's efficient identity-based signatures [35] and Boneh-Lynn-Sham's short signature scheme [8]. The results are shown in Table 2. Although the computation time for the signer key generation is much higher than the reference schemes, the computation time of signature generation and verification are similar to the reference schemes for the 1024 Bits of Dlog security. It shows that our schemes are practical for the real world applications. Meanwhile, for the higher security requirement, our scheme shows that our signature generation computation time is just about 17% more than Paterson-Schuldt's IDS scheme compare with the benefits form our schemes, it is acceptable. In the Verification computation time, the security setting for our scheme was set at 12 levels of security and the verifier policy was set at 5, which is about in the middle of the security levels. Hence, the cost of computation was about double of Paterson-Schuldt's IDS scheme for the 2048 Bits of Dlog security.

6 Conclusion

Privacy issue over the information shared in the organization without an efficient and proper control mechanism has motivated us to provide schemes to resolve it. The notion of a multi-level controlled signature scheme captures the need for the integrity, authenticity, and authority, which presents as a perfect tool to enable access control systems for a large organization where the hierarchical structure are applied. To further enhancing the privacy of our scheme by hiding the security-level policy, it can achieved by simply removing the security level policy from the hash in the first part of a signature ($\sigma_1 = h(R||pk_S||pk_{TA}||M)$).

However, there are some remaining issues that can be handled in the future work. For the credential revokable property in the event of the credential disputes or discloses to others, a system should be able to resolve without reissuing a new credential to other users.

References

[1] "PBC Library," http://crypto.stanford.edu/pbc [Online; accessed on September 10, 2019].

[2] W. Bagga and R. Molva, "Policy-based cryptography and applications," in *Proc. of the 9th International Conference on Financial Cryptography and Data Security (FC'05), Roseau, The Commonwealth of Dominica*, ser. Lecture Notes in Computer Science, vol. 3570. Springer-Verlag, February-March 2005, pp. 72–87.

[3] M. Bellare and G. Fuchsbauer, "Policy-based signatures," in *Proc. of the 17th International Conference on Practice and Theory in Public-Key Cryptograph (PKC'14), Buenos Aires, Argentina*, ser. Lecture Notes in Computer Science, vol. 8383. Springer-Verlag, March 2014, pp. 520–537.

[4] M. Bellare and G. Neven, "Multi-signatures in the plain public-key model and a general forking lemma," in *Proc. of the 13th ACM Conference on Computer and Communications Security (CCS'06), Alexandria, Virginia, USA*. ACM, October-November 2006, pp. 390–399.

[5] D. Boneh, "The decision diffie-hellman problem," in *Proc. of the 3rd International Symposium on Algorithmic Number Theory (ANTS'98), Portland, Oregon, USA*, ser. Lecture Notes in Computer Science, vol. 1423. Springer-Verlag, June 1998, pp. 48–63.

[6] D. Boneh, X. Boyen, and E.-J. Goh, "Hierarchical identity based encryption with constant size ciphertext," in *Proc. of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'05), Aarhus, Denmark*, ser. Lecture Notes in Computer Science, vol. 3494. Springer-Verlag, May 2005, pp. 440–456.

[7] D. Boneh and M. K. Franklin, "Identity-based encryption from the weil pairing," in *Proc. of the 21st Annual International Cryptology Conference (CRYPTO'01), Santa Barbara, California, USA*, ser. Lecture Notes in Computer Science, vol. 2139. Springer-Verlag, August 2001, pp. 213–229.

[8] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in *Proc. of the 7th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'01), Gold Coast, Australia*, ser. Lecture Notes in Computer Science, vol. 2248. Springer-Verlag, December 2001, pp. 514–532.

[9] X. Boyen and B. Waters, "Anonymous hierarchical identity-based encryption (without random oracles)," in *Proc. of the 26th Annual International Cryptology Conference (CRYPTO'06), Santa Barbara, California, USA*, ser. Lecture Notes in Computer Science, vol. 4117. Springer-Verlag, August 2006, pp. 290–307.

[10] A. D. Caro and V. Iovino, "jpbc: Java pairing based cryptography," in *Proc. of the 16th IEEE Symposium on Computers and Communications (ISCC'11), Kerkyra, Corfu, Greece*. IEEE, July 2011, pp. 850–855.

[11] P. Chen, Y. Wu, J. Su, and X. Wang, "Comparing performance of hierarchical identity-based signature schemes," *IEICE Transactions*, vol. 99-D, no. 12, pp. 3181–3184, 2016.

[12] S. S. M. Chow, L. C. K. Hui, S.-M. Yiu, and K. P. Chow, "Secure hierarchical identity based signature and its application," in *Proc. of the 6th International Conference on Information and Communications Security (ICICS'04), Malaga, Spain*, ser. Lecture Notes in Computer Science, vol. 3269. Springer-Verlag, October 2004, pp. 480–494.

[13] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Information Theory*, vol. 22, no. 6, pp. 644–654, November 1976.

[14] A. Escala, J. Herranz, and P. Morillo, "Revocable attribute-based signatures with adaptive security in the standard model," in *Proc. of the 4th International Conference on Cryptology in Africa (AFRICACRYPT'11), Dakar, Senegal*, ser. Lecture Notes in Computer Science, vol. 6737. Springer-Verlag, July 2011, pp. 224–241.

[15] C.-I. Fan, C.-N. Wu, W.-K. Chen, and W.-Z. Sun, "Attribute-based strong designated-verifier signature scheme," *Journal of Systems and Software*, vol. 85, no. 4, pp. 944–959, April 2012.

[16] C. Gentry and A. Silverberg, "Hierarchical id-based cryptography," in *Proc. of the 8th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'02), Queenstown, New Zealand*, ser. Lecture Notes in Computer Science, vol. 2501. Springer-Verlag, December 2002, pp. 548–566.

[17] J. Herranz, "Attribute-based signatures from rsa," *Theoretical Computer Science*, vol. 527, pp. 73–82, March 2014.

[18] J. Herranz, F. Laguillaumie, B. Libert, and C. Ràfols, "Short attribute-based signatures for threshold predicates," in *Proc. of the Cryptographer's Track of the RSA Conference (CT-RSA'12), San Francisco, CA, USA*, ser. Lecture Notes in Computer Science, vol. 7178. Springer-Verlag, February-March 2012, pp. 51–67.

[19] J. Horwitz and B. Lynn, "Toward hierarchical identity-based encryption," in *Proc. of the 21th International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands*, ser. Lecture Notes in Computer Science, vol. 2332. Springer-Verlag, April-May 2002, pp. 466–481.

[20] S. Hou, X. Huang, J. K. Liu, J. Li, and L. Xu, "Universal designated verifier transitive signatures for graph-based big data," *Information Sciences*, vol. 318, pp. 144–156, October 2015.

[21] X. Huang, Y. Mu, W. Susilo, and F. Zhang, "Short designated verifier proxy signature from pairings," in *Proc. of the Workshops on Embedded and Ubiquitous Computing (EUC'05), Nagasaki, Japan*, ser. Lecture Notes in Computer Science, vol. 3823. Springer-Verlag, December 2005, pp. 835–844.

[22] X. Huang, W. Susilo, Y. Mu, and F. Zhang, "Short (identity-based) strong designated verifier signature schemes," in *Proc. of the 2nd International Conference on Information Security Practice and Experience (ISPEC'06), Hangzhou, China*, ser. Lecture Notes in Computer Science, vol. 3903. Springer-Verlag, April 2006, pp. 214–225.

[23] M. Jakobsson, K. Sako, and R. Impagliazzo, "Designated verifier proofs and their applications," in *Proc. of the International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT'96), Saragossa, Spain*, ser. Lecture Notes in Computer Science, vol. 1070. Springer-Verlag, May 1996, pp. 143–154.

[24] F. Laguillaumie and D. Vergnaud, "Designated verifier signatures: Anonymity and efficient construction from any bilinear map," in *Proc. of the 4th International Conference on Security in Communication Networks (SCN'04), Amalfi, Italy*, ser. Lecture Notes in Computer Science, vol. 3352. Springer-Verlag, September 2004, pp. 105–119.

[25] F. Laguillaumie and D. Vergnaud, "Multi-designated verifiers signatures: anonymity without encryption," *Information Processing Letters*, vol. 102, no. 2-3, pp. 127–132, April 2007.

[26] J. Li, M. H. Au, W. Susilo, D. Xie, and K. Ren, "Attribute-based signature and its applications," in *Proc. of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS'10), Beijing, China*. ACM, April 2010, pp. 60–69.

[27] Y. Li, H. Lipmaa, and D. Pei, "On delegatability of four designated verifier signatures," in *Proc. of the 7th International Conference on Information and Communications Security (ICICS'05), Beijing, China*, ser. Lecture Notes in Computer Science, vol. 3783. Springer-Verlag, December 2005, pp. 61–71.

[28] H. Lipmaa, G. Wang, and F. Bao, "Designated verifier signature schemes: Attacks, new security notions and a new construction," in *Proc. of the 32nd International Colloquium on Automata, Languages and Programming*

(ICALP'05), Lisbon, Portugal, ser. Lecture Notes in Computer Science, vol. 3580. Springer-Verlag, July 2005, pp. 459–471.

[29] M. Mahmoody and A. Mohammed, "On the power of hierarchical identity-based encryption," in *Proc. of the 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'16), Vienna, Austria*, ser. Lecture Notes in Computer Science, vol. 9666. Springer-Verlag, May 2016, pp. 243–272.

[30] H. K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-based signatures: Achieving attribute-privacy and collusion-resistance," *IACR Cryptology ePrint Archive*, vol. 2008, p. 328, 2008.

[31] H. K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-based signatures," in *Proc. of the Conference on the Cryptographers' Track at the RSA (CT-RSA'11), San Francisco, CA, USA*, ser. Lecture Notes in Computer Science, vol. 6558. Springer-Verlag, February 2011, pp. 376–392.

[32] T. Okamoto and K. Takashima, "Efficient attribute-based signatures for non-monotone predicates in the standard model," in *Proc. of the 14th International Conference on Practice and Theory in Public Key Cryptography (PKC'11), Taormina, Italy*, ser. Lecture Notes in Computer Science, vol. 6571. Springer-Verlag, March 2011, pp. 35–52.

[33] T. Okamoto and K. Takashima, "Decentralized attribute-based signatures," in *Proc. of the 16th International Conference on Practice and Theory in Public-Key Cryptography (PKC'13), Nara, Japan*, ser. Lecture Notes in Computer Science, vol. 7778. Springer-Verlag, February-March 2013, pp. 125–142.

[34] T. Okamoto and K. Takashima, "Efficient attribute-based signatures for non-monotone predicates in the standard model," *IEEE Transactions on Cloud Computing*, vol. 2, no. 4, pp. 409–421, October-December 2014.

[35] K. G. Paterson and J. C. N. Schuldt, "Efficient identity-based signatures secure in the standard model," in *Proc. of the 11th Australasian Conference on Information Security and Privacy (ACISP'06), Melbourne, Australia*, ser. Lecture Notes in Computer Science, vol. 4058. Springer-Verlag, July 2006, pp. 207–222.

[36] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *Journal of Cryptology*, vol. 13, no. 3, pp. 361–396, 2000.

[37] S. Saeednia, S. Kremer, and O. Markowitch, "An efficient strong designated verifier signature scheme," in *Proc. of the 6th International Conference on Information Security and Cryptology (ICISC'03), Seoul, Korea*, ser. Lecture Notes in Computer Science, vol. 2971. Springer-Verlag, November 2003, pp. 40–54.

[38] Y. Sakai, N. Attrapadung, and G. Hanaoka, "Attribute-based signatures for circuits from bilinear map," in *Proc. of the 19th IACR International Conference on Practice and Theory in Public-Key Cryptography (PKC'16), Taipei, Taiwan*, ser. Lecture Notes in Computer Science, vol. 9614. Springer-Verlag, March 2016, pp. 283–300.

[39] S. F. Shahandashti and R. Safavi-Naini, "Threshold attribute-based signatures and their application to anonymous credential systems," in *Proc. of the 2nd International Conference on Cryptology in Africa (AFRICACRYPT'09), Gammarth, Tunisia*, ser. Lecture Notes in Computer Science, vol. 5580. Springer-Verlag, June 2009, pp. 198–216.

[40] A. Shamir, "Identity-based cryptosystems and signature schemes." in *Proc. of the 4th Annual International Cryptology Conference (CRYPTO'84), Santa Barbara, California, USA*, ser. Lecture Notes in Computer Science, vol. 196. Springer-Verlag, August 1984, pp. 47–53.

[41] W. Susilo, F. Zhang, and Y. Mu, "Identity-based strong designated verifier signature schemes," in *Proc. of the 9th Australasian Conference on Information Security and Privacy (ACISP'04), Sydney, Australia*, ser. Lecture Notes in Computer Science, vol. 3108. Springer-Verlag, July 2004, pp. 313–324.

[42] P. Thorncharoensri, W. Susilo, and Y. Mu, "Multi-level controlled signature," in *Proc. of the 13th International Workshop on Information Security Applications (WISA'12), Jeju Island, Korea*, ser. Lecture Notes in Computer Science, vol. 7690. Springer-Verlag, August 2012, pp. 96–110.

[43] P. Thorncharoensri, W. Susilo, and Y. Mu, "Policy-controlled signatures and their applications," *Computer Standards & Interfaces*, vol. 50, pp. 26–41, 2017.

[44] R. Tso, T. Okamoto, and E. Okamoto, "Practical strong designated verifier signature schemes based on double discrete logarithms," in *Proc. of the 1st SKLOIS Conference on Information Security and Cryptology (CISC'05), Beijing, China*, ser. Lecture Notes in Computer Science, vol. 3822. Springer-Verlag, December 2005, pp. 113–127.

[45] L. Zhang, Y. Mu, and Q. Wu, “Compact anonymous hierarchical identity-based encryption with constant size private keys,” *The Computer Journal*, vol. 59, no. 4, pp. 452–461, April 2016.

Author Biography

Pairat Thorncharoensri is a Lecturer in the School of Computer Science and Information Technology and a member of the Institute of Cybersecurity and Cryptology, University of Wollongong (UOW), Australia. He received his Master of Computer Science and Doctor of Philosophy degree from University of Wollongong. He received his Bachelor Degree in Electrical Engineering from King Mongkut’s University of Technology North Bangkok, Thailand. His main research interest is currently focused on privacy preservative technique in Internet of things, Blockchain, cloud computing and big data.

Willy Susilo is a Senior Professor in the School of Computing and Information Technology, Faculty of Engineering and Information Sciences in University of Wollongong, Australia. He is the director of Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong. Willy is an innovative educator and researcher. Currently, he is the Head of School of Computing and Information Technology at UOW (2015 - now). Prior to this role, he was awarded the prestigious Australian Research Council Future Fellowship in 2009. He was the former Head of School of Computer Science and Software Engineering (2009 - 2010) and the Deputy Director of ICT Research Institute at UOW (2006 - 2008). He is the Editor in Chief of the Information journal. Willy obtained his PhD from the University of Wollongong in 2001. He has published more than 300 papers in journals and conference proceedings in cryptography and network security. He has served as the program committee member of several international conferences. In 2016, he was awarded the researcher of the Year?at UOW, due to his research excellence and contributions. His work on the creation of short signature schemes has been well cited and it is part of the IETF draft.

Joonsang Baek is a Senior Lecturer in the School of Computer Science and Information Technology and a member of the Institute of Cybersecurity and Cryptology, University of Wollongong (UOW), Australia. He was a Research Scientist in the Institute for Infocomm Research, Singapore, and an Assistant Professor in the Khalifa University of Science and Technology, United Arab Emirates. Joonsang received his PhD from Monash University, Australia, in 2004. His PhD thesis was on security analysis of signcryption, and has received great attention from the research community. He has published his work in numerous reputable journals and conference proceedings. His current research interests are in the field of applied cryptography and cybersecurity. He has also served as a Program Committee Member and the Chair for a number of renowned conferences on information security and cryptography.