Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 8736531, 19 pages
https://doi.org/10.1155/2022/8736531

Research Article

Secure and Efficient Communication in VANETs Using Level-Based Access Control

P. Thorncharoensri, W. Susilo, and Y. Chow

Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Wollongong, NSW 2522, Australia

Correspondence should be addressed to P. Thorncharoensri; pairat@uow.edu.au

Received 2 September 2021; Revised 6 December 2021; Accepted 22 December 2021; Published 29 March 2022

Academic Editor: Iftikhar Ahmad

In recent years, the development of vehicular ad-hoc networks (VANETs) has received much attention in intelligent transportation systems (ITS). Unlike traditional ad-hoc networks, VANETs are emerging with unique characteristics that share similar technology with autonomous vehicles (AVs) and automated driving systems (ASDs). Communication between vehicles and the surrounding infrastructure unit, such as a roadside unit (RSU), must be secured, concise, and authentic. Hence, an access control system for the ad-hoc environment is required. We introduced a level-based controlled signcryption (LBS) scheme, which can be easily constructed and implemented into an access control system for VANETs environment. Our encrypted message has a short and constant size, which is better when compared with other attribute-based signcryption or encryption. Confidentiality, privacy, and authenticity are also provided in our scheme to ensure secure and authentic communication. Therefore, our scheme has addressed communication cost, scalability, security, and privacy issues in VANETs. This primitive can be applied to simplify attribute-based access control, as the only attribute required is an integer representing the security level. Our objective is to improve the quality and security of VANET communication. Moreover, an optional privacy mechanism in our scheme provides flexibility in controlling node privacy in VANETs.

1. Introduction

Vehicular ad-hoc networks (VANETs) are a key enabling technology in the development of autonomous vehicle technology and intelligent transportation systems (ITS). A VANET is a form of communication network that connects vehicles-to-vehicles and vehicles to roadside infrastructure. For example, Figure 1 shows communication between roadside units (RSUs) and vehicles. It also shows that vehicles can access information at the edge of an ITS server or the Internet. VANETs together with ITS and the Internet deliver a wide range of services, such as route guidance, traffic conditions (e.g., average vehicle density and speed), safety alerts, proximity advertisements, location-based positioning information, and entertainment. In general, vehicles in VANETs need to connect wirelessly to ITS applications and the Internet via RSUs. Vehicles exchange information with edge ITS servers to access a variety of local environmental data. For example, a subject's location positioning data are obtained through real-time point cloud data from LiDAR devices and cameras. This crucial information is particularly important for vehicles to be aware of any hidden out-of-sight vehicles/subjects that cannot be observed by the vehicle itself [1].

This is the fundamental information distribution system for autonomous vehicles (AVs). Despite the fact that VANETs and AV networks share similarities in communication and security issues, research in both areas have typically been conducted separately [2]. Secure broadcast communication plays a significant role in both systems. Whether using purely cellular, purely ad-hoc, or a hybrid architecture [3], VANETs need a confidential, efficient, concise, and precise communication protocol and access control.

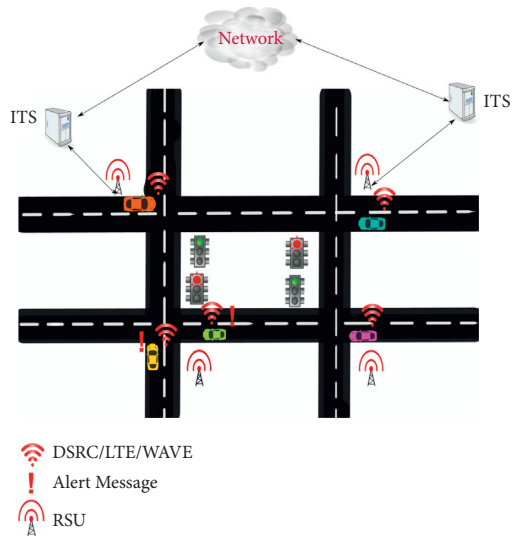

Figure 1: VANETs.

To date, many security mechanisms have been proposed for VANETs. Insight on security issues related to VANETs can be found in surveys conducted by Malhi et al. [4] and Engoulou et al. [5]. The security challenges described in Malhi et al. [4] are the dependence on infrastructure, RSU range of communication, high mobility, difficulty in trust management, huge amount of data, scalability, and high cost. Our work aims to provide an efficient primitive for access control and secure communication, which will solve many of the security challenges related to the communication aspect of VANETs. For example, an efficient primitive should be able to work efficiently in architectures for high mobility, and it should provide a high degree of scalability for secure network access and have low communication overhead to reduce the amount of data in the network. An attribute-based access control where an integer representing the security level is the only attribute in the system will help simplify the hierarchical structure of a VANET. We call this level-based access control. The purpose of clustering protocols in VANETs aims to improve the performance of target tracking, traffic estimation, misbehaviour detection, privacy preservation, and certificate revocation, which makes significant contribution toward the dependence on infrastructure issue [6]. By providing an optional privacy mechanism in our primitive, it is easy to construct secure revocable local cluster communication.

Other than secure, fast, and efficient communication in VANETs, authentication, integrity, confidentiality, access control, and privacy are security requirements that are important in VANETs. Nevertheless, security requirements are application dependent. For example, message broadcasts from an RSU and message broadcasts from/to neighbouring vehicles may need to use two different primitives to satisfy VANET security requirements. To emphasise the importance of broadcast communication in VANETs, a survey on the quality-of-service (QoS) in relation to broadcast protocols is summarised [7]. It is an unavoidable fact that broadcast protocols play a significant role in VANETs for delivering emergency messages for the safety of vehicles or for reducing ad-hoc communication overhead. Due to the nature of VANETs that comprise of transient nodes and often unexpected changes in network topology, level-based access control can significantly improve broadcast performance.

Let us illustrate this based on the following example. The Bell–LaPadula model can be used for simple but efficient access control in sharing resources and communication in a VANET (or ad-hoc network) environment. However, based on the aforementioned reason, we argue that multilevel security can help to make communication in VANETs simple but secure. For example, let us consider a VANET with a number of security levels, such as top-security level, high-security level, operator security level, and general user level. An ambulance, which is a node in the VANET, wants a roadside unit to arrange the best route to a hospital. The roadside unit, which has the operator security level, only needs to know that the ambulance is a high-security level node in order for it to help manage the route and traffic. Assuming that all vehicles on the road are autonomous vehicles, the roadside unit may request that other vehicles make way for the ambulance due to its high-security level.

1.1. Related Work. There are three areas of work that closely relate to our work. These are reviewed in this section.

First, there is attribute-based and policy-based cryptography. Sahai and Waters were the first to propose attribute-based encryption in [8]. However, the implementation is not flexible and is only limited to particular systems. Later, Goyal et al. formalised and demonstrated a more general key-policy design for attribute-based encryption in [9]. An attribute-based encryption (ABE) scheme is a cryptographic primitive that only allows users who satisfy the fine-grained access structure to decrypt the ciphertext. A ciphertext is generated by encrypting a message along with a public string p. By using a decryption key associated with a Boolean predicate function $\varphi(.)$, the ciphertext can be decrypted only when $\varphi(p)$ is true. Their ABE scheme can be classified as a key-policy attribute-based encryption (KP-ABE). Basic KP-ABE uses threshold gates to express the access policy.

In later work on ABE schemes, logic gates (AND/OR) and threshold gates were used to construct the policy in ABE. A ciphertext-policy attribute-based encryption (CP-ABE) scheme was first proposed by Bethencourt et al. [10]. This primitive allows the policy to be constructed with a combination of numerical comparison and logic gates. Attribute-based access control benefited from this primitive in practice. Hence, their work is used for comparison with our work in a later section. Following their work, much research on CP-ABE has been published within the last decade. Recently, a CP-ABE scheme for a complex constructed attribute with multiauthorities was proposed by Rouselakis and Waters [11]. Their scheme supports any string for the attribute in the policy, and it can combine multiple trusted authorities in a single policy. This is an efficient and practical scheme that can cover a wide range of users, on the condition that the attributes and the policy must not be complicated. Hence, this work was also included for comparison with our scheme. Another CP-ABE scheme,

which is constructed without a bilinear function, was proposed by Yao and others [12]. Their scheme achieves significantly fast computation, which is suitable in the Internet of thing (IoT) and wireless sensor network (WSN). Therefore, our work is also compared with this primitive in the comparison section of this study. Moreover, Xue et al. proposed a comparable attribute-based encryption scheme (CABE) in [13]. The CABE scheme provides a new way for attribute comparison which is an inspiration for our work. It allows the system to test whether an attribute is equal to, more than, or less than the given policy.

Similar to attribute-based cryptography, Bagga and Molva [14] introduced the notion of policy-based cryptography (policy-based encryption and policy-based signature schemes). The policy in CP-ABE proposed by Bethencourt et al. [10] is conceptually similar to policy-based encryption. However, the access structure and complexity of the policy are more comprehensive, which can easily be used in an access control system. Due to the nature of applications in our work, we only applied the concept of a policy tree implementing integer comparison. With only an integer attribute in the policy (no AND/OR logic gates), our scheme achieves a constant size ciphertext.

Bellare and Fuchsbauer later formally defined the security definition of policy-based signatures [15]. Like the attribute-based signature, a signer in the policy-based signature can sign on a message only if he/she meets the policy's requirement (or attributes). From a signer's point of view, both attribute-based and policy-based signatures ensure message integrity and authenticity properties. The authenticity and integrity of the message can be validated by only a verifier. Nevertheless, the signature cannot be generated or forged by other beside the signer who met the policy requirements. A variant policy-controlled signature scheme was proposed in [16], where authenticity was done in a reversed manner and limited to only the attribute-credential holder (verifier). In contrast to attribute-based and policy-based signatures, policy-controlled signatures guarantee that only verifiers, who hold credentials that satisfy the policy or attributes, can verify the authenticity and integrity of the message and the nonreputability of the signer.

Closely related work on an attribute-based signature (ABS) scheme was presented in 2008 by Maji et al. [17]. They also proposed an ABS scheme in the standard model in [18]. In this primitive, a signer reveals nothing about their identity in the signing process. It shows only their specific attributes. Many variant ABS schemes were put forward based on Maji et al.'s works. These are presented in [19–26]. Shahandashti and Safavi-Naini [19] and Li et al. [27] independently published their work on the same topic of ABS with threshold predicate. Escala and others [22] also presented an ABS scheme with threshold predicate. Their scheme was equipped with a revocation method. They also formalised the adaptive unforgeability property for their ABS scheme. Additionally, a constant size ABS with threshold predicate was proposed by Herranz et al. [21]. Later, an ABS scheme with arbitrary circuits, which is supposed to be more efficient than the ABS schemes proposed by Maji et al. [18, 25], was presented by Sakai et al. [26]. This is true only when we increase the gate numbers. Their construction is based on the combination of a witness indistinguishable, an extractable noninteractive proof system and an existentially unforgeable signature scheme.

As mentioned above, many ABE and ABS schemes were proposed in the last decade. It seems that a simple solution for the aforementioned problems can easily be achieved by applying attribute-based or policy-based cryptography to encrypt a message and its signature. Only qualified users, who hold the proper credentials, can recover the message from an encrypted message. However, this is not entirely true since authentication of the signature can be transferred to a third party. Since the products after the decryption process are the signature and the message, they can be verified by any party. From the aforementioned scenario, signcryption's ability to authenticate and decrypt should be limited to verifiers who hold credentials that satisfy the security level.

The second related topics is the hierarchical identity-based signature and encryption scheme (HIBS and HIBE). HIBE is a concept that combines an identity-based encryption (IBE) scheme and a hierarchical system. In the hierarchical system, a user's identity at the security level k is able to generate a secret key for the identity of its descendants. It also can decrypt a ciphertext meant for its descendants but not for other identities. Hence, it can provide a similar solution to the aforementioned problems about multilevel security. Our work has a slightly similar aspect; however, it differs in key generation and distribution. For example, every level in our scheme only has a single branch, but it can generate multiple private keys for the branch. Following the introduction of HIBE, there has been much subsequent work in this area [28, 29]. As a natural conversion from the HIBE scheme, the hierarchical identity-based signature (HIBS) scheme also inherits its properties. The ancestor's identity can generate a secret key for the identity of its descendants, and it can generate the signature of behalf of its descendants as well. However, it cannot generate a signature on a message on behalf of other identities.

Boneh et al. was the first to propose a HIBE scheme with a constant size ciphertext in [30]. Our work on a constant size ciphertext was inspired by their work. Zhang et al. [28] proposed a compact size HIBE with a constant size private key. Their scheme's distinct feature is having a constant size for both private key and ciphertext. Our scheme achieves this feature as well. A review of existing HIBS schemes can be found in [29]. Chen et al. provided a performance comparison between four HIBS schemes. Our work uses a similar approach for comparison with various schemes. In general, the benefit of using HIBE and HIBS in the network system that applied the identity-based public key system is to mitigate the bottleneck in the communication. Moreover, it is also to limit the key escrow. Nevertheless, similar to ABS schemes, HIBS only provides authenticity and message integrity for a signer, but not nonrepudiation and authorisation for a verifier.

The final related area of work is signcryption and attribute-based signcryption. Zheng was the first to proposed

the signcryption scheme in [31]. This primitive provides both authenticity and secrecy. The first attribute-based signcryption (ABSC) scheme with threshold predicate was presented by Gagné et al. [32]. The primitive is a solution to the problems mentioned earlier, but is much less efficient when compared to our scheme in term of communication overhead and computation. Following the above work, several researchers proposed some works [33–35] to improve ABSC schemes in terms of applications and concrete construction in the standard model, efficient constructions, and formal security models, respectively. Pandit et al. [33] provided an ABSC scheme in a standard model. Their scheme proved to be secure in the IND-CCA in the adaptive-predicate model. The ABSC-based linear codes secret sharing scheme was introduced by Song et al. in [34]. The access control system based on ABSC scheme was proposed by Zheng et al. in [35]. They showed that the key agreement scheme, an important tool for the access control system, can be constructed efficiently with ABSC. Rao [36] proposed an attribute-based online/offline signcryption scheme, which reduced the computation signing time when needed. A certificateless threshold signcryption scheme proposed by Yu et al. in [37] allows t-out-of-n senders to work together to generate the signcryption for applications such as petition, voting. Wang et al. [38] proposed an efficient signcryption with a designated equality test. It allows a trusted third party to check the validity of ciphertext with a given message. Later on, Le et al. put forward a signcryption scheme in the standard model in [39], which adopted the concept of the designated equality test by Wang et al. in [38]. Nevertheless, in Wang et al.'s scheme [38], a signature on a message can easily be extracted from a signcrypted ciphertext. Hence, a signcrypted ciphertext can be linked to the sender identity. Recently, Yu et al. [40] introduced a lightweight hybrid ABSC scheme for IoT systems. Their schemes allow a low-power and low-computation device to pass the heavy computation in the signing and encrypting process to the edge node before returning the partial output. However, its communication cost (ciphertext size) is linear to the attributes; hence, it is only suitable for a simple access control system for the IoT system. The features comparison between signcryption, HIBE, ABE, ABSC, and our LBS schemes are given in Table 1.

Finally, the access control schemes for ad-hoc networks or the Internet of Things are reviewed. There are many access control systems that have been proposed recently [41–44]. Vijayakumar et al. [41] proposed an access control scheme that can be secure against attacks such as message reply attack, Sybil attack, masquerading attack, integrity attacks, and collusion attacks. Wazid et al. [42] proposed an access control scheme that is secured in the fog computing environment. Recently, Xia et al. [43] introduced an efficient authentication and key agreement scheme for a secure and anonymous access control system. It allows users to identify themselves without revealing their identity. This anonymous property is intriguing the optional anonymity in our scheme. Our scheme is designed to easily apply with the access control system, which allows users to securely and privately transmit their broadcast messages or private data to other users. It also reduces the process of encryption and signing in the existing access control schemes mentioned above. To further the benefits from privacy-preserving in our scheme, we believe that our scheme can integrate with AI-enabled blockchain-based access control proposed by Bera [44]. Hence, the anonymity in intrusion detection systems that detect and mitigate malicious attacks for the Internet of Everything (IOE) environment can be constructed. Moreover, our scheme can be designed for the access control system in an ad-hoc environment, such as VANETs, since a node can prove itself to another node that it holds a valid credential. Moreover, the proof cannot be replicated and transferred to other parties. Hence, it provides another privacy protection layer on top of it.

1.2. Our Contributions. To tackle the secrecy and privacy issue in communication over VANETs or AV networks, we propose a level-based controlled signcryption (LBS) scheme for level-based access control systems. This primitive provides access control with a hierarchical structure or a simple multilevel access control. Level-based access control can be viewed as a simplified version of attribute-based access control where the only attribute is an integer representing the security level. Our aim is to flexibly and efficiently allow nodes to communicate with other nodes securely and privately. Hence, we provide an optional privacy mechanism in the signcrypting (sign with encrypt) and unsigncrypting (verify with decrypt) processes. The security of this primitive is also formalised in this study. The notion of LBS schemes allows receivers, who hold a credential of a certain security level specified by the sender, to verify the authenticity of the ciphertext and to decrypt it. Moreover, with optional privacy, higher levels of confidentiality can be provided in the LBS scheme. LBS with optional privacy provides privacy between two or more parties when required. Nodes in VANETs can securely, confidentially, and flexibly send messages to an individual node, a group of node, or all nodes (message broadcasts are often use in VANETs to get the current routing in the network). Hence, the LBS scheme is suitable for mass encryption with authenticity and privacy.

We organized our study as follows. Some preliminaries and definitions are defined in the next section. In Section 3, the formal definition of level-based controlled signcryption is presented. The security model is described as well in the same section. The construction detail of LBS scheme is described with its security proofs in Section 4. Finally, a comparison of our concrete scheme with ABS schemes and the conclusion of this study are presented in the last two sections.

2. Preliminaries

2.1. Notation. The rest of this study will use the following notations. We said that a function $(f: \mathbb{N} \longrightarrow \mathbb{R})$ is negligible, if $\forall c > 0$, and for all sufficiently large n, $f(n) < 1/n^c$. poly(.) is a deterministic polynomial function. Let $[n]$ represent a series of numbers (or indexes), e g., if n is integer, then $[n] = \{0, \ldots, n\}$. Let poly be a polynomial function. The

Table 1: Comparison of encryptions schemes with our LBS scheme.

Schemes \ properties	Multi receivers	Confidentiality	Authenticity	Privacy	Constant size
YW21(SignCryption) [37]	×	✓	✓	×	✓
LDR21(SignCryption) [39]	×	✓	✓	×	✓
ZMW16(HIBE) [28]	✓	✓	×	×	✓
GPSW06(KP-ABE) [9]	✓	✓	×	×	×
XHXWYH17(CABE) [13]	✓	✓	×	×	×
BSW07(CP-ABE) [10]	✓	✓	✓	×	×
RW15(CP-ABE) [11]	✓	✓	×	×	×
YTC14(CP-ABE) [12]	✓	✓	×	×	×
PPB16(ABSC) [33]	✓	✓	✓	✓	×
SLLL17(ABSC) [34]	✓	✓	✓	✓	×
Rao17(ABSC) [36]	✓	✓	✓	✓	×
YLWXY20(ABSC) [40]	✓	✓	✓	✓	×
Our LBS	✓	✓	✓	×	✓
Our LBS with OP	✓	✓	✓	✓	✓

security parameter is denoted by ℓ, and the polynomial time is denoted by $\mathscr{P}$. We say that $\mathscr{P} \in \text{poly}(1^{\ell})$ if $\forall \text{poly}(1^{\ell}); \forall \ell$: $\mathscr{P} \leq \text{poly}(1^{\ell})$.

A process that randomly selects the element l from a set L is denoted by $l \xleftarrow{\$} L$. Let H: $\{0,1\}^* \longrightarrow \mathbb{G}_1$ be a collision-resistant hash function that maps a string to $\mathbb{G}_1$. Let h: $\{0,1\}^* \longrightarrow \mathbb{Z}_p^*$ be a collision-resistant hash function that maps a string to $\mathbb{Z}_p^*$.

2.2. Bilinear Pairing. Let us define $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ cyclic multiplicative groups. The generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ groups are g_1 and g_2, respectively. p is defined as a prime number and the order of all generators. $\hat{e}$ is denoted as an efficient algorithm that maps $\mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$. We call this function a bilinear mapping function.

The bilinear mapping function has 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) Nondegeneracy: $\exists g_1 \in \mathbb{G}_1 \exists g_2 \in \mathbb{G}_2$: $\hat{e}(g_1, g_2) \neq 1$

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

$\varphi(.)$ is defined as an existing function mapping $\mathbb{G}_1$ to $\mathbb{G}_2$ in a one-time unit. It is also true in another way around.

2.3. Complexity Assumptions

Definition 1 (Computational Diffie–Hellman (CDH) problem). With a triple $(\mathbf{g}, \mathbf{g}^{\chi}, \mathbf{g}^{\psi} \in \mathbb{G}_1)$ as input and $g^{\chi \cdot \psi}$ as output, we said that an algorithm $\mathscr{A}$ with an advantage probability ϵ' breaks the CDH problem if

$$\Pr\left[\mathscr{A}(g, g^{\chi}, g^{\psi}) = g^{\chi \cdot \psi}\right] \geq \varepsilon. \tag{1}$$

Note that the probability is taken over the random bits used by $\mathscr{A}$ and the arbitrarily selected random integers $\chi, \psi \in \mathbb{Z}_q^*$.

Assumption 1. Computational Diffie–Hellman assumption.

The (t, ϵ')-CDH assumption is intact if the probabilistic polynomial time-bound (PPT) algorithm with time complexity $t(.)$ and a probability advantage of at least ϵ' breaking the CDH problem does not exist.

Definition 2 (Decisional bilinear Diffie–Hellman (DBDH) problem). With a random quadruple $(\mathbf{g}, \mathbf{g^x}, \mathbf{g^y}, \mathbf{g^z}) \in \mathbb{G}_1$ and a random integer $\mathbf{Z} \in \mathbb{G}_T$ as input, determine if $\mathbf{Z} = \hat{e}(\mathbf{g}, \mathbf{g})^{\mathbf{x} \cdot \mathbf{y} \cdot \mathbf{z}}$ or not. An algorithm $\mathscr{A}$ solves the DBDH problem in $\mathbb{G}_1, \mathbb{G}_T$ within t, ϵ', if $\mathscr{A}$ runs in time t, and

$$\begin{aligned} &\left|\Pr\left[\mathscr{A}(\mathrm{g}, \mathrm{g}^{\mathrm{x}}, \mathrm{g}^{\mathrm{y}}, \mathrm{g}^{\mathrm{z}}, \mathbf{Z} = \hat{e}(\mathrm{g}, \mathrm{g})^{\mathrm{x} \cdot \mathrm{y} \cdot \mathrm{z}}) = 1\right]\right. \\ &\left. -\Pr\left[\mathscr{A}\left(\mathrm{g}, \mathrm{g}^{\mathrm{x}}, \mathrm{g}^{\mathrm{y}}, \mathrm{g}^{\mathrm{z}}, \mathbf{Z} = \hat{e}(\mathrm{g}, \mathrm{g})^{\mathrm{d}}\right) = 1\right]\right| \geq \epsilon'. \end{aligned} \tag{2}$$

Note that the above probability is taken over the random bits used by $\mathscr{A}$, random integers $\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{d} \in \mathbb{Z}_p$, a random group $g \in \mathbb{G}_1$.

Assumption 2. Decisional bilinear Diffie–Hellman assumption.

The (t, ϵ')-DBDH assumption in $\mathbb{G}_1, \mathbb{G}_T$ holds only if no PPT algorithm (t, ϵ') solving the DBDH problem does not exist.

3. Level-Based Signcryption Schemes with Optional Privacy (LBS)

There are three main players in the level-based signcryption (LBS) scheme. A sender S generates a ciphertext that can only be verified and decrypted by a receiver V who holds a credential that satisfies the level-based security policy. This scenario can be used to ensure communication among authorised users. It can be used for message broadcasts in VANETs. It can also be used in secure group chats. However, if a sender wants to limit verification of the ciphertext to a particular receiver, the sender needs to place a personal receiver's public key in the signcryption. We called this "optional privacy." The "optional privacy" scenario is useful for sending private messages between the sender and the receiver. This solution can also be applied to a group of receivers where they share a group secret key, which is not

affected by a member's secret key. The last player is a trusted authority TA. In this level-based security system, a credential is generated by TA, where it associated the credential with the given security level.

Notations on security level and level-based policy are defined as follows: L is a security level that specified in the level-based security policy. LP is defined as a level-based security policy that contains a security level as a security clearance of a verifier. Let us consider the following example. LP = "L > = l" means l is the minimum security level, and the policy is set such that l is the minimum level that can decrypt the ciphertext. Note that other types of index or symbol, such as A, B, C, and D, also can be used to indicate the security level in LP. We assume that the security levels are in ascending order, which means that the higher the number, the higher the security clearance.

An LBS scheme is a sextuple (Setup, TKeyGen, SKeyGen, CreGen, SCrypt, Verify).

System parameter generation (Setup): with a security parameter ℓ as input, Setup, a probabilistic function, generates the system parameter param as follows:

$$\text{Setup}\left(1^{\ell}\right) \longrightarrow \text{param}. \tag{3}$$

TA key generator (TKeyGen): With param as input, TKeyGen, a probabilistic function, generates the private key (sk_{TA}) and the public parameter (pk_{TA}) of a trusted authority as follows:

$$\text{TKeyGen}\left(\text{param}\right) \longrightarrow \left(\text{pk}_{\text{TA}}, \text{sk}_{\text{TA}}\right). \tag{4}$$

Signer key generator (SKeyGen): with param and pk_{TA} as input, SKeyGen, a probabilistic function, generates the private key (sk_E) and the public parameter (pk_E) of a signer as follows:

$$\text{SKeyGen}\left(\text{param}, \text{pk}_{\text{TA}}\right) \longrightarrow \left(\text{pk}_E, \text{sk}_E\right). \tag{5}$$

Verifier credential generator (CreGen): with param, sk_{TA}, and an assertion L indicating the security level of a verifier as input, CreGen, a probabilistic function, generates a verifier's credential $\mathbb{C}$ as follows:

$$\text{CreGen}\left(\text{param}, \text{sk}_{\text{TA}}, \text{L}\right) \longrightarrow \mathbb{C}. \tag{6}$$

Signcryption (SCrypt): with param, pk_{TA}, sk_E, pk_E, a message M, and a level-based security policy LP as input, SCrypt, a probabilistic function, generates a sender's ciphertext $\mathscr{C}$, that is,

$$\text{SCrypt}\left(\text{param}, M, \text{sk}_E, \text{pk}_E, \text{pk}_{\text{TA}}, \text{LP}\right) \longrightarrow \mathscr{C}. \tag{7}$$

Unsigncryption (USCrypt): with param, pk_{TA}, pk_E, LP, $\mathbb{C}$, and C as input, USCrypt, a deterministic function, returns with the decisional output $d \in \{M, \text{reject}\}$, that is,

$$\text{USCrypt}\left(\text{param}, \mathscr{C}, \text{pk}_{\text{TA}}, \text{pk}_E, \text{LP}, \mathbb{C}\right) \longrightarrow d. \tag{8}$$

Signcryption with optional privacy (SCOP): with param, pk_{TA}, sk_E, pk_E, pk_R, a message M, and a level-based security policy LP as input, SCrypt, a probabilistic function, generates a sender's ciphertext $\mathscr{C}$ as follows:

$$\text{SCOP}\left(\text{param}, M, \text{sk}_E, \text{pk}_E, \text{pk}_R, \text{pk}_{TA}, \text{LP}\right) \longrightarrow \mathscr{C}. \tag{9}$$

Unsigncryption with optional privacy (USCOP): with param, pk_{TA}, pk_E, sk_R, LP, $\mathbb{C}$, and $\mathscr{C}$ as input, USCOP, a deterministic function, determines the validity of the input and returns the decisional output $d \in \{M, \text{reject}\}$, that is,

$$\text{USCOP}\left(\text{param}, \mathscr{C}, \text{pk}_{\text{TA}}, \text{pk}_E, \text{sk}_R, \text{LP}, \mathbb{C}\right) \longrightarrow d. \tag{10}$$

3.1. Unforgeability. The unforgeability property in LBS provides an assurance that an attacker allowed to access the credential queries that cannot generate a valid level-based signcryption $\mathscr{C}^*$, which seems to be signed by the sender S on a new message M^*. The attacker is allowed to access the signing oracle **SSO** and the verifying oracle **VCO**. The pk_E and pk_{TA} is also known to the attacker; however, a signer's secret key sk_E is restricted. Even the attacker can arbitrarily select a message M, the entire credentials, and a level-based security level policy LP as input, a valid signcryption on a new message M^* cannot be generated. We called this a security against existential unforgeability under the adaptive chosen message and credentials exposure attack (EUF – CMCEA).

Let us define some notations before given a formal definition. Let $\mathscr{A}_U$ be the adaptively chosen message and credentials exposure adversary. We assume that $\mathscr{A}_U$ is an adversary who attacks the unforgeability of the LBS scheme. $\mathscr{F}$ is defined as a simulator who run the attack simulation with $\mathscr{A}_U$. Note that $\mathbb{C}$ is defined as the credentials of the entire security level, for example, if the system has 15 security levels, then $\mathbb{C} = (\mathbb{C}_1, \ldots, \mathbb{C}_{15})$. The **SSO** and **VCO** oracles that describe the ability of adversaries to break the unforgeability of an LBS scheme are illustrated as follows.

SSO(.,.): at most o_S times, when a query for a signcryption $\mathscr{C}$ on its choice of message M and a signer S corresponding to pk_E is issued, **SSO** runs the Sign algorithm to generate a signcryption $\mathscr{C}$ on a message M corresponding to pk_{TA}, pk_E, and LP. Finally, **SSO** responses to the query with $\mathscr{C}$.

VCO(.): at most o_C times, when a query for credential $\mathbb{C}_i$ corresponding to the arbitrarily chosen security level L is issued, **VCO** responses with the corresponding credentials $\mathbb{C}$.

The formalisation of unforgeability is now defined in the following statements. For a level-based signcryption, an adversary $\mathscr{A}_U$ is associated with the experiment given in Algorithm 1. $\mathscr{A}_U$ has two functions, namely, st_R stage and forge stage. Using an adaptive strategy, $\mathscr{A}_U$, in the st_R stage, arbitrarily chooses a message and makes queries to the signing oracle **SSO**(.,.) and the credential oracle **VCO**(.). The query processes are allowed repeatedly according to $\mathscr{A}_U$'s strategies. At the end of this stage, $\mathscr{A}_U$ outputs a message M and a level-based security policy LP along with some state information (st) to be used in the forge state. In the second stage, $\mathscr{A}_U$ takes M, LP, st as input and outputs a

Input: A security parameter k
Output: The success of the attack.
(1) param$\xleftarrow{\$}$Setup$(1^k)$
(2) $(\mathrm{pk}_{\mathrm{TA}}, \mathrm{sk}_{\mathrm{TA}})\xleftarrow{\$}$TKeyGen(param)
(3) $(\mathrm{pk}_E, \mathrm{sk}_E)\xleftarrow{\$}$SKeyGen(param, $\mathrm{pk}_{\mathrm{TA}}$)
(4) Phase I: learning
(5) $(\mathrm{st}, M, \mathtt{LP})\xleftarrow{\$}\mathscr{A}_U^{\mathbf{SSO}(.,.),\mathbf{VCO}(.)}(\mathrm{st}_R, \mathrm{pk}_{\mathrm{TA}}, \mathrm{pk}_E, \mathrm{param})$
(6) Phase II: challenging
(7) $C\xleftarrow{\$}\mathscr{A}_U^{\mathbf{VCO}(.)}(\mathrm{forge}, \mathrm{st}, M, \mathtt{LP})$
(8) If **sso**$(M, \mathtt{LP})$ has never been executed;USCrypt(param, C, $\mathrm{pk}_{\mathrm{TA}}$, pk_E,   </p> LP, $\mathbb{C}$) = M then
(9) return *1*
(10) else
(11) return 0
(12) end

Algorithm 1: Algorithm for unforgeability experiment $(\mathrm{Expt}_{\mathscr{A}_{\mathrm{EUF},\Sigma}}^{\mathrm{EUF-CMACE}})$.

valid level-based signcryption $\mathscr{C}$. $\mathscr{A}_U$ wins the above experiment if

(1) $\mathscr{A}_U$ results a forged signcryption δ on a new message M^* corresponding to pk_E and LP
(2) $M^*\leftarrow\mathrm{USCrypt}(\mathscr{C}, \mathrm{pk}_E, \mathtt{LP}, \mathbb{C})$
(3) $\mathscr{A}_U$ never makes a request for a level-based signcryption with $M^*, \mathrm{pk}_E, \mathtt{LP}$ to the **SSO** oracle

$\mathrm{ADV}_{\mathrm{EUF-CMCEA}}(.)$ denotes the probability that $\mathscr{A}_U$ successfully passes the winning condition in the above simulation.

Definition 3. Under a chosen message and credentials exposure attack model, an LBS scheme is $(\mathrm{t}, o_S, o_C, \epsilon)$-secure existential unforgeability, if the PPT adversary $\mathscr{A}_U$ success in the attack the nonnegligible (within k) probability

$$\mathrm{ADV}_{\mathrm{EUF-CMCEA}}(\mathrm{k}) = \Pr\left[\mathrm{Expt}_{\mathscr{A}_U,\Sigma}^{\mathrm{EUF-CMCEA}}(\mathrm{k}) = 1\right] = \epsilon, \tag{11}$$

does not exist. Let $\mathscr{A}_U$ runs at most t times and makes at most o_S signing queries and o_C credential queries.

3.2. Indistinguishability. The indistinguishable property is modelled on the indistinguishability of ciphertext in the selective-credential exposed model. The attack models can be divided into two different models, which are the chosen plaintext attack and the (active) chosen ciphertext attack. The first attack model aims to prevent a group of corrupted credential holders (malicious receivers) from verifying and decrypting a level-based ciphertext C on a message M with a level-based security policy LP, where these malicious receivers do not have enough credentials to satisfy the security level indicated in LP. In the second attack model, an attacker has the added power to query for verification of any signature because excepted for the challenge signature, he/she does not have credentials that satisfy the level-based security policy LP to verify it by himself/herself.

Before describing the formal definition of these models, some definitions are first defined. $\mathscr{A}_I$ is defined as the adaptive chosen plaintext (or ciphertext) and chosen level-based security policy distinguisher.

3.2.1. Chosen Plaintext Attack (IND-CPA). The signcryption oracle **SSO**, credential generator oracle **VCO**, and verification oracle **VSO** are used to describe the abilities of $\mathscr{A}_I$ breaking the chosen plaintext attack in the selective-credential exposed model (IND – CPA). The **VCO** and **SSO** oracles are described as follows:

VCO(.): at most o_C times, when a query for the credential $\mathbb{C}_i$ corresponding to the arbitrarily chosen security level L is issued, **VCO** responses with the corresponding credentials $\mathbb{C}$.

(i) **SSO**(.,.): at most o_E times, $\mathscr{A}_I$ can arbitrarily choose a message M and make a query for a ciphertext $\mathscr{C}$. **SSO** responses with the result of the SCrypt algorithm generating a ciphertext $\mathscr{C}$ corresponding to a message M, $\mathrm{pk}_{\mathrm{TA}}$, pk_E, and LP.

Formalisation of indistinguishability under the adaptive chosen plaintext attack model of an LBS scheme is described as follows. The experiment that is associated with an adversary $\mathscr{A}_I$ of a level-based signcryption scheme is illustrated in Algorithm 2. $\mathscr{A}_I$ has two functions that are st_C stage and guess stage. Using an adaptive strategy, in the st_C stage, $\mathscr{A}_I$ arbitrarily chooses a message and makes queries to the signing oracle **SSO**(.,.) and the credential oracle **VCO**(.). The query processes are allowed repeatedly according to $\mathscr{A}_I$'s strategies. $\mathscr{A}_I$ outputs two messages (M_0, M_1), a level-based security policy LP and some state information (st) used in the guess state at the end of the stage. Based on a previously set bit b, the experiment runs SCrypt with a message M_b as an input and outputs a valid level-based ciphertext $\mathscr{C}$. In the second stage (Phase 2), $\mathscr{A}_I$ takes $M_0, M_1, \mathtt{LP}, \mathrm{st}, \mathscr{C}$ as input and outputs a guess bit (1 or 0). Note that $\mathscr{A}_I$ still can make queries to the **VCO** oracle before outputting a guess, and if the following conditions are not satisfied, the experiment will be aborted.

(1) $\mathscr{A}_I$ never issues a query for a level-based ciphertext with LP and M as input to the **SSO** oracle

Input: A security parameter k and a random bit b
Output: The success of the attack.
(1) Setup parameters:
(2) $\text{param} \xleftarrow{\$} \text{Setup}(1^k)$
(3) $(\text{pk}_{\text{TA}}, \text{sk}_{\text{TA}}) \xleftarrow{\$} \text{TKeyGen}(\text{param})$
(4) $(\text{pk}_E, \text{sk}_E) \xleftarrow{\$} \text{SKeyGen}(\text{param}, \text{pk}_{\text{TA}})$
(5) Phase I: learning
$(\text{st}, M_0, M_1, \text{LP}) \xleftarrow{\$} \mathscr{A}_I^{\text{SSO}(.),\text{VCO}(.)}(\text{st}_C, \text{pk}_{TA}, \text{pk}_E, \text{param})$;
(6) Prechallenge:
If $b = 0$, then
$\text{C} \xleftarrow{\$} \text{SCrypt}(\text{param}, M_0, \text{sk}_E, \text{pk}_E, \text{pk}_{\text{TA}}, \text{LP})$
(7) else,
$\text{C} \xleftarrow{\$} \text{SCrypt}(\text{param}, M_1, \text{sk}_E, \text{pk}_E, \text{pk}_{\text{TA}}, \text{LP})$
end
(8) Phase II: challenging
$d \xleftarrow{\$} \mathscr{A}_I^{\text{VCO}(.)}(\text{guess}, \text{st}, M_0, M_1, \text{LP}, \text{C})$
If $\text{vco}(\text{L}: \text{L} > = \min(\text{LP}))$ and $\text{sso}(M_{[b]}, \text{L})$ have never been executed, then
(9) If
$d = b$, then return *1*
else return *0*
else
return $\perp$
end

Algorithm 2: Algorithm for indistinguishability experiment $(\text{Expt}_{\mathscr{A}_I,\Sigma}^{\text{IND-CPA}(b)})$.

(2) With a restriction that $\text{LP} = \text{“L} \geq l\text{”}$, $\mathscr{A}_I$ can only make a request for credentials with a security level $\text{L} < l$ to the **VCO** oracle

Let $\text{ADV}_{\text{IND-CPA}}(.)$ be the success probability of $\mathscr{A}_I$ breaking the indistinguishability under the adaptive chosen plaintext attack in the selective-credential exposed model of an LBS scheme.

Definition 4. An LBS scheme is $(\text{t}, o_E, o_C, \epsilon)$-secure indistinguishability under a chosen plaintext attack in the selective-credential exposed model if the PPT distinguisher $\mathscr{A}_I$ success in the attack with a nonnegligible (within k) probability

$$\text{ADV}_{\text{IND-CPA}}(\text{k}) = \left|\Pr\left[\text{Expt}_{\mathscr{A}_I,\Sigma}^{\text{IND-CPA}(0)}(\text{k})\right] - \Pr\left[\text{Expt}_{\mathscr{A}_I,\Sigma}^{\text{IND-CPA}(0)}(\text{k})\right]\right| = \epsilon, \tag{12}$$

does not exist. Note that $\mathscr{A}_I$ runs at most t times and makes at most o_E signcryption queries and o_C credential queries.

3.2.2. Chosen Ciphertext Attack (IND-CCA). Before formalising this attack, we first give some definitions and outline the oracles. Let (IND – CPA) be indistinguishability under a chosen plaintext attack in the selective-credential exposed model. The signing oracle **SSO**, credential generator oracle **VCO**, and unsigncryption oracle **USO** are used to describe the abilities of $\mathscr{A}_C$ breaking (IND – CPA) of an LBS scheme. The **SSO** and **VCO** oracles have been depicted in Section 3.2.1. The **USO** oracle, however, is described as follows.

USO(.,.): with a ciphertext $\mathscr{C}$ and a level-based security policy LP as input, $\mathscr{A}_I$ can request up to o_D queries for the decryption and verification of a ciphertext $\mathscr{C}$. **USO** returns a decisional result d that is either reject regrading the authenticity of ciphertext $\mathscr{C}$ or a message M.

Formalisation of indistinguishability under the adaptive chosen ciphertext attack model of an LBS scheme is described as follows. The experiment in Algorithm 3 depicts an adversary $\mathscr{A}_I$ of a level-based signcryption scheme. There are two functions that are a st_C stage and a guess stage, which $\mathscr{A}_I$ can execute. In the first stage, $\mathscr{A}_I$, with some elastic and pliable strategies, chooses a message and issues a request to the credential queries **VCO**(.), the signing queries **SSO**(.,.), and the unsigncryption queries **USO**(.,.). The query processes are allowed repeatedly according to $\mathscr{A}_I$'s strategies. $\mathscr{A}_I$ outputs two messages (M_0, M_1), a level-based security policy LP, and some state information (st) used in the guess state at the end of the stage. The experiment runs SCrypt with a message M_b as an input and outputs a valid level-based ciphertext $\mathscr{C}$. In the guess stage (Phase 2), $\mathscr{A}_I$ takes $M_0, M_1, \text{LP}, \text{st}, \mathscr{C}$ as input and outputs a guess bit (1 or 0). Note that $\mathscr{A}_I$ can still make queries to **VCO** and **USO** oracles before outputting a guess. However, if the following conditions are not satisfied, the experiment will be aborted.

(1) $\mathscr{A}_I$ never issues a query for a level-based ciphertext with LP and M_0 or M_1 as input to the **SSO** oracle
(2) $\mathscr{A}_I$ never issues a query with $\mathscr{C}$ as input to the **USO** oracle
(3) With a restriction that $\text{LP} = \text{“L} \geq l\text{”}$, $\mathscr{A}_I$ can only make a request for credentials with a security level $\text{L} < l$ to the **VCO** oracle

Let $\text{ADV}_{\text{IND-CCA}}(.)$ be the success probability of $\mathscr{A}_I$ breaking indistinguishability under the adaptive chosen

Input: a security parameter k and a random bit b
Output: the success of the attack.
Setup parameters: the setup process is same with Algorithm 2.
Phase I: learning
$(\mathrm{st}, M_0, M_1, \mathrm{LP}) \xleftarrow{\$} \mathscr{A}_I^{\mathbf{SSO}(.),\mathbf{VCO}(.),\mathbf{USO}(.)}(\mathrm{st}_C, \mathrm{pk}_{TA}, \mathrm{pk}_E, \mathrm{param})$
Prechallenge: the prechallenge process is same with Algorithm 2.
(1) Phase II: challenging
$d \xleftarrow{\$} \mathscr{A}_I^{\mathbf{SSO}(.),\mathbf{VCO}(.),\mathbf{USO}(.)}(\mathrm{guess}, \mathrm{st}, M_0, M_1, \mathrm{LP}, \mathrm{C})$ if
$\mathrm{vco}(\mathrm{L}: \mathrm{L} > = \min(\mathrm{LP}))$ and
$\mathbf{SSO}(M_{[b]}, \mathrm{L})$ and
$\mathbf{USO}(\mathrm{C}, .)$ have never been executed, then
If $d = b$, then return *1*
else return *0*
else
return $\perp$
end

Algorithm 3: Algorithm for indistinguishability experiment under CCA attack ($\mathrm{Expt}_{\mathscr{A}_I,\Sigma}^{\mathrm{IND-CCA}(b)}$).

ciphertext attack in the selective-credential exposed model of an LBS scheme.

Definition 5. An LBS scheme is $(\mathrm{t}, o_E, o_C, o_D, \epsilon)$-secure indistinguishability of ciphertext under a chosen ciphertext attack in the selective-credential exposed model if a distinguisher $\mathscr{A}_I$ success in the attack with a nonnegligible (within k) probability

$$\mathrm{ADV}_{\mathrm{IND-CCA}}(k) = \left|\Pr\left[\mathrm{Expt}_{\mathscr{A}_I,\Sigma}^{\mathrm{IND-CCA}(0)}(k)\right] - \Pr\left[\mathrm{Expt}_{\mathscr{A}_I,\Sigma}^{\mathrm{IND-CCA}(1)}(k)\right]\right| = \epsilon, \tag{13}$$

does not exist. Note that $\mathscr{A}_I$ runs at most t times and makes at most o_E signcryption queries, o_C credential queries, and o_D unsigncryption queries.

4. The Level-Based Signcryption (LBS) Scheme

In this section, we present our concrete construction of the LBS scheme. The scheme is described as follows.

Setup: with ℓ as the input, Setup randomly selects a prime $p \approx \mathrm{poly}(1^\ell)$. Setup select three groups $\mathbb{G}_1$, $\mathbb{G}_1$, and $\mathbb{G}_T$ of the prime order p. Setup constructs a bilinear mapping function $\widehat{e}$ that maps $\mathbb{G}_1$ and $\mathbb{G}_2$ to $\mathbb{G}_T$. The above mapping function is defined as $\widehat{e}$: $\mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$. Choose a random generator $g \in \mathbb{G}_1$, $o \in \mathbb{G}_2$. Construct a hash function $h(.)$, such that h: $\{0,1\}^* \longrightarrow \mathbb{Z}_p$. Finally, the system parameter param is composed of $(p, \widehat{e}, g, o, h)$. param is returned as a response of the queries.

TKeyGen: the total security level is denoted by n. Given param as input, TKeyGen computes a private key $\mathrm{sk}_{\mathrm{TA}}$ and a public key $\mathrm{pk}_{\mathrm{TA}}$ randomly for each security level as follows: choose random integers $\mu_0, \ldots \mu_n, \gamma_0, \ldots, \gamma_n, a, b \in \mathbb{Z}_p$. Let

$$\mathrm{pk}_{TA} = \left(U_1 = o^{\mu_1 \cdot a}, \ldots U_n = o^{\mu_n \cdot a}, V_0 = g^{\gamma_0}, \ldots, V_n = g^{\gamma_n}, W_1 = g^{\sum_{i=1}^{n} \gamma_1 \cdot \mu_i \cdot a + \gamma_0 \cdot \mu_0 \cdot a + b}, \ldots, W_n = g^{\gamma_n \cdot \mu_n \cdot a + \gamma_0 \cdot \mu_0 \cdot a + b}\right). \tag{14}$$

denote a public key. Then, TKeyGen returns $\mathrm{sk}_{\mathrm{TA}} = (\mu_0, \ldots, \mu_n, \gamma_0, \ldots, \gamma_n, a, b)$, as the trusted authority's private key and $\mathrm{pk}_{\mathrm{TA}} = (\mathbb{U}, \mathbb{V}, \mathbb{W})$ as the trusted authority's public key. Note that $\mathbb{U}$, $\mathbb{V}$, and $\mathbb{W}$ are the symbols that represent the set of vectors $(U_1, \ldots, U_n)$, $(V_0, \ldots, V_n)$, and $(W_1, \ldots, W_n)$, respectively.

SKeyGen: on input of a system parameter param and a public key of the trusted authority, SKeyGen randomly generates a private key sk_E and a public key pk_E as follows: first, choose a random integer $x \in \mathbb{Z}_p$. Let $\mathbb{X} = (X_0 = V_0^x, \ldots, X_n = V_n^x)$. Let $sk_E = x$ be the signer's private key and $\mathrm{pk}_E = \mathbb{X}$ be the signer's public key. At the end, SKeyGen outputs $\mathrm{sk}_E, \mathrm{pk}_E$.

CreGen: L_{L} denotes a list that contains the generated credentials. A security level of a verifier, for example, $\mathrm{L} =$ 'D″ or "5" is denoted by L. Given param, $\mathrm{sk}_{\mathrm{TA}}$, $\mathrm{pk}_{\mathrm{TA}}$, and $\mathrm{L} = l$, CreGen randomly generates a credential csk_E with the following processes. CreGen arbitrary chooses $\nu_0 \in \mathbb{Z}_p^*$: $\nu_0 \notin \mathrm{L}_{\mathrm{L}}$. Then, CreGen generates a credential at a security level of $\mathrm{L} = l$ by computed $\nu_l = ((\gamma_0 \cdot \mu_0 + \gamma_l \cdot \mu_l) \cdot a + b - \nu_0 \cdot \gamma_0)/\gamma_l$; $C_0 = o^{\nu_0}$; $C_l = o^{\nu_l}$. CreGen stores $\mathbb{C}_V = (C_0, C_l)$ in L_{L} and returns $\mathbb{C}_V$ as a credential of $\mathrm{L} = l$ to the verifier.

SCrypt: given param, $\mathrm{pk}_{\mathrm{TA}}$, sk_E, pk_E, $\mathrm{LP} =$ "$\mathrm{L} \geq l$" and a message M, an encryptor E computes a ciphertext C as follows:

$$r \leftarrow^{\$} \mathbb{Z}_p,\ R = \hat{e}(W_l, o)^r, \sigma_1 = h\left(R \| \mathrm{pk}_E \| \mathrm{pk}_{\mathrm{TA}} \| \mathtt{LP} \| \sigma_3\right), \quad \sigma_2 = o^{\sigma_1 \cdot x + r}, \sigma_3 = R \oplus M. \tag{15}$$

The level-based ciphertext is $\mathscr{C} = (\sigma_1, \sigma_2, \sigma_3)$. S publishes $\sigma, \mathtt{LP}$.

USCrypt: a receiver D possesses a credential $\mathtt{L} = k$, where $k \geq l$. Given pk_E, pk_{TA}, $\mathbb{C}$, $\mathtt{LP}$ = "$\mathtt{L} \geq l$", σ decrypt $\mathscr{C}$ as follows:

(1) Compute

$$R = \hat{e}(W_l, \sigma_2) \cdot \hat{e}(X_0, C_0)^{-\sigma_1} \cdot \hat{e}(X_k, C_k)^{-\sigma_1} \cdot \prod_{i=l; i \neq k}^{n} \hat{e}(X_i, U_i)^{-\sigma_1}. \tag{16}$$

(2) Check whether or not the following equation holds.

$$\sigma_1 \stackrel{?}{=} h\left(R \| \mathrm{pk}_E \| \mathrm{pk}_{\mathrm{TA}} \| \mathtt{LP} \| \sigma_3\right). \tag{17}$$

(3) If the equation in (1) holds, then decrypt the ciphertext as follows.

$$M = \sigma_3 \oplus R. \tag{18}$$

SCOP: in optional privacy, the security policy needs to be set to the lowest level to cover all possible users, since the ciphertext is intended for a single receiver R or a group with a shared secret key. Given param, $\mathrm{pk}_{\mathrm{TA}}$, sk_E, pk_E, pk_R, $\mathbb{C}_E$, $\mathtt{LP}$ = "$\mathtt{L} \geq 1$", and a message M, an encryptor E who possesses a credential for a security level assertion $\mathtt{L} = k$, where $k \geq 1$, computes a ciphertext $\mathscr{C}$ as follows:

$$\begin{aligned} &r \leftarrow^{\$} \mathbb{Z}_p, R = \hat{e}(W_l, o)^r \\ &\widehat{R} = \left(\hat{e}\left(X_{R,0}, C_{E,0}\right) \cdot \hat{e}\left(X_{R,k}, C_{E,k}\right) \prod_{i=1; i \neq k}^{n} \hat{e}\left(X_{R,i}, U_i\right)\right)^{h(\sigma_3)} \\ &\sigma_3 = R \oplus M, \overline{R} = X_{R,0}^{x_E \cdot h(\sigma_3)} \\ &\sigma_1 = h\left(\|\widehat{R}\| \overline{R} \| R \| \mathrm{pk}_E \| \mathrm{pk}_R \| \mathrm{pk}_{\mathrm{TA}} \| \mathtt{LP} \| \sigma_3\right) \\ &\sigma_2 = o^{\sigma_1 \cdot x + r}. \end{aligned} \tag{19}$$

The level-based ciphertext is $\mathscr{C} = (\sigma_1, \sigma_2, \sigma_3)$. S publishes $\sigma, \mathtt{LP}$.

USCOP: a receiver D who possesses a credential for a security level assertion $\mathtt{L} = k$, where $k \geq l$. Given $\mathrm{sk}_R, \mathrm{pk}_E$, pk_{TA}, $\mathbb{C}_R$, $\mathtt{LP}$ = "$\mathtt{L} \geq l$", σ decrypts $\mathscr{C}$ as follows:

(1) Compute

$$\begin{aligned} &R = \hat{e}(W_l, \sigma_2) \cdot \hat{e}\left(X_{R,0}, C_{R,0}\right)^{-\sigma_1} \cdot \hat{e}\left(X_{R,k}, C_{R,k}\right)^{-\sigma_1} \prod_{i=l; i \neq k}^{n} \hat{e}(X_i, U_i)^{-\sigma_1}, \\ &\overline{R} = X_{E,0}^{x_R \cdot h(\sigma_3)}, \\ &\widehat{R} = \left(\hat{e}\left(X_{E,0}, C_{R,0}\right) \cdot \hat{e}\left(X_{E,k}, C_{R,k}\right) \prod_{i=1; i \neq k}^{n} \hat{e}\left(X_{E,i}, U_i\right)\right)^{h(\sigma_3))}. \end{aligned} \tag{20}$$

(2) Check whether or not the following equation holds.

$$\sigma_1 \stackrel{?}{=} h\left(\widehat{R} \| \overline{R} \| R \| \mathrm{pk}_E \| \mathrm{pk}_R \| \mathrm{pk}_{\mathrm{TA}} \| \mathtt{LP} \| \sigma_3\right). \tag{21}$$

(3) If the equation in (1) holds, then decrypt the ciphertext as follows.

$$M = \sigma_3 \oplus R. \tag{22}$$

Note that $\hat{e}(W_l, o)$, $\hat{e}(X_{R,0}, C_{R,0})$, $\hat{e}(X_{R,k}, C_{R,k})$, $\hat{e}(X_{R,i}, U_i))$, $\hat{e}(X_{R,0}, C_{E,0})$, $\hat{e}X_{R,k}, C_{E,k})$, and $\hat{e}(X_{R,i}, U_i))$ are precomputed and can be reused in the signcryption or signcryption process of other signcryption. Hence, only one pairing computation is needed in the unsigncryption algorithm. The optional privacy process has similar computation costs, where the computation of the bilinear paring component in $R, \widehat{R}$, and $\overline{R}$ can be precomputed once and saved or reused later.

4.1. Completeness. The signcryption and unsigncryption algorithms can be verified its completeness with the following equations.

$$
\begin{aligned}
&\text{Generated by the sender} = \text{generated by the receiver}\\
&\widehat{e}(W_l,o)^r = \prod_{i=l;i\neq k}^{n}\widehat{e}\left(g^{\gamma_i\cdot x},o^{\mu_i\cdot a}\right)^{-\sigma_1}\\
&\quad=\widehat{e}(W_l,o)^r\cdot\widehat{e}(W_l,o)^{\sigma_1\cdot x}\cdot\left(\widehat{e}(g,o)^{\gamma_0\cdot\nu_0}\cdot\widehat{e}(g,o)^{\left((\gamma_0\cdot\mu_0+\gamma_k\cdot\mu_k)\cdot a+b-\nu_0\cdot\gamma_0\right)}\right.\\
&\quad=\widehat{e}(W_l,o^{\sigma_1\cdot x+r})\cdot\widehat{e}\left(g^{\gamma_0\cdot x},o^{\nu_0}\right)^{-\sigma_1}\cdot\widehat{e}\left(g^{\gamma_k\cdot x},o^{\left((\gamma_0\cdot\mu_0+\gamma_k\cdot\mu_k)\cdot a+b-\nu_0\cdot\gamma_0\right)/\gamma_k}\right)^{-\sigma_1}\\
&\quad=\widehat{e}(W_l,\sigma_2)\cdot\widehat{e}(X_0,C_0)^{-\sigma_1}\cdot\widehat{e}(X_k,C_k)^{-\sigma_1}\\
&\qquad\cdot\prod_{i=l;i\neq k}^{n}\widehat{e}(X_i,U_i)^{-\sigma_1}\cdot\widehat{e}(g,o)^{\sum_{i=l;i\neq k}^{n}\gamma_i\cdot\mu_i\cdot a)^{-\sigma_1\cdot x}}\\
&\quad=\widehat{e}(W_l,o)^r\cdot\widehat{e}\left(g^{\sum_{i=l}^{n}\gamma_i\cdot\mu_i\cdot a+\gamma_0\cdot\mu_0\cdot a+b},o\right)^{\sigma_1\cdot x}\cdot\widehat{e}(g,o)^{-\sigma_1\cdot x\cdot\left(\sum_{i=l}^{n}\gamma_i\cdot\mu_i\cdot a+\gamma_0\cdot\mu_0\cdot a+b\right)}.\\
&\widehat{e}(W_l,o)^r=\widehat{e}(W_l,o)^r.
\end{aligned}
\tag{23}
$$

4.2. Unforgeability

Theorem 1. *Under the adaptive chosen message and credential exposure attack model, the level-based signcryption scheme is existentially unforgeable if and only if the CDH assumption is intact under the random oracle model.*

Proof. We start with the assumption that a forger algorithm $\mathscr{A}_U$ that can win the existential unforgeability model described in Section 3.1 exists. Using this $\mathscr{A}_U$, we can construct an adversary algorithm $\mathscr{F}$ to break 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, sets $g=g_1, o=g_1^y$, and obtains param $=(p,\widehat{e},g,o,h)$. $\mathscr{F}$ then runs TKeyGen to obtain TA's public-private keys and sets $\mathbb{X}=(X_0=g_1^{x\cdot\gamma_0},\ldots,X_n=g_1^{x\cdot\gamma_n})$ as the signer's public key pk_E. The following algorithms are the construction of queries that are later used in the simulation.

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

VCO queries: given $\mathrm{sk}_{\mathrm{TA}}$ as input, **VCO** executes CreGen for the credential VCR, where $\mathtt{L}=l$. **VCO** outputs VCR.

SSO queries: given "$\mathtt{LP}=\mathtt{L}\geq l$" and a message M as input, **SSO** generated a level-based signcryption with the following equations.

$$
\begin{aligned}
&r,\iota_1,\iota_2\xleftarrow{\$}\mathbb{Z}_p,K\xleftarrow{\$}\mathbb{G}_T,\\
&\quad R=\widehat{e}(W_l^x,o)^{\iota_1}\widehat{e}(W_l,o^r):\ W_l^x=\left(g_1^x\right)^{\sum_{i=l}^{n}\gamma_i\cdot\mu_i\cdot a+\gamma_0\cdot\mu_0\cdot a+b},\\
&\quad\sigma_1=h\left(R\|\mathrm{pk}_E\|\mathrm{pk}_{\mathrm{TA}}\|\mathtt{LP}\|\sigma_3\right)=\iota_1,\\
&\quad\sigma_2=o^r,\sigma_3=R\oplus M,\\
&\quad\Gamma=R\|\mathrm{pk}_E\|\mathrm{pk}_{\mathrm{TA}}\|\mathtt{LP}\|\sigma_3.
\end{aligned}
\tag{24}
$$

Note that **SSO** has access to the list (Γ,ι) via $\mathscr{F}$. **SSO** uses this advantage to update (Γ,ι) to the list in **HO**. **SSO** then responds with $\mathscr{C}=(\sigma_1,\sigma_2,\sigma_3)$.

Optional privacy: let $\mathbb{X}_R=(X_{R,0}=g_1^{x_R\cdot\gamma_0},\ldots,X_{R,n}=g_1^{x_R\cdot\gamma_n})$ as the receiver's public key pk_R. **SSO** generated a level-based signcryption with the following equations.

$$
\begin{aligned}
&r,\iota_1,\iota_2,\iota_3\xleftarrow{\$}\mathbb{Z}_p,K\xleftarrow{\$}\mathbb{G}_T\\
&\quad R=\widehat{e}(W_l^x,o)^{\iota_1}\widehat{e}(W_l,o^r):\ W_l^x=\left(g_1^x\right)^{\sum_{i=l}^{n}\gamma_i\cdot\mu_i\cdot a+\gamma_0\cdot\mu_0\cdot a+b}\\
&\quad\widehat{R}=\left(\widehat{e}(W_1^x,o)\right)^{h(\sigma_3)}\\
&\quad h(\sigma_3)=\iota_3\\
&\quad\sigma_1=h\left(\widehat{R}\|\overline{R}\|R\|\mathrm{pk}_E\|\mathrm{pk}_R\|\mathrm{pk}_{\mathrm{TA}}\|\mathtt{LP}\|\sigma_3\right)=\iota_1\\
&\quad\sigma_2=o^r,\sigma_3=R\oplus M\\
&\quad\Gamma=\widehat{R}\|\overline{R}\|R\|\mathrm{pk}_E\|\mathrm{pk}_R\|\mathrm{pk}_{\mathrm{TA}}\|\mathtt{LP}\|\sigma_3.
\end{aligned}
\tag{25}
$$

Note that $\overline{R}$ does not need to be computed. It only needs to be verified by the following equation when $\mathscr{A}_U$ submits the query of the hash value for σ_1.

$$\widehat{e}(\overline{R}, g) \stackrel{?}{=} \widehat{e}\left(X_{R,0}, X_{E,0}\right)^{\iota_3}. \tag{26}$$

If it matches with $X_{R,0}$ in the list of verifier public key and the list of queried ciphertext, it then returns the corresponding result. Otherwise, it returns the new ι_1.

We start the simulation by allowing $\mathscr{A}_U$ to access the above queries. In fact, $\mathscr{A}_U$ always issues a request of query to **HO** queries before outputting the forgery. Let us denote this potential forgery by $M^*, \mathscr{C}^*, \mathtt{LP}^*$. With any adaptive strategy, $\mathscr{A}_U$ interacts with the simulator and eventually outputs a forgery.

$\mathscr{A}_U$ successfully completes the challenge, if, with a message M^* and $\mathtt{LP}^*$ as input, $\mathscr{A}_U$ generates a valid signcryption $\mathscr{C}^*$. Note that this forgery should not be a result of **SSO** queries.

The success probability $\mathrm{ADV}_{\mathrm{EUF-CMCEA}}(.)$ of $\mathscr{A}_U$ winning the simulation is denoted by ϵ. Let the base of the natural logarithm be denoted by e. In the random oracle, the input of every query made for a signcryption is a result of a query to **HO** oracle. Hence, $q_H \geq q_S$. Let solve the CHD problem using the forgery generated by $\mathscr{A}_U$. The forking technique in [45, 46] is applied in the following technique for the probability analysis.

First, a signcryption $\mathscr{C}^*$ on message M^* where $\sigma_1^* = h(\Gamma^*) = \iota^*$ is obtained in the first round.

Then, $\mathscr{F}$ resets $\mathscr{A}_U$ to the initial state and repeats the above simulation again. Note that in the second simulation, $\mathscr{A}_U$ uses a different hash value $\sigma_1' = h(\Gamma^*) = \iota'$.

Therefore, $\mathscr{A}_U$ eventually should generate another signcryption $\mathscr{C}'$ on message M^* where $h(\Gamma^*) = \iota\prime$.

With these two signcryptions, $\mathscr{F}$ computes

$$\begin{aligned}
\left(\frac{\sigma_2^*}{\sigma_2'}\right)^{1/\sigma_1^* - \sigma_1'} &= \left(\frac{o^{\sigma_1^* \cdot x + r}}{o^{\sigma_1 \prime \cdot x + r}}\right)^{1/\sigma_1^* - \sigma_1'} \\
&= \left(\frac{\left(g_1^y\right)^{\iota^* \cdot x + r}}{\left(g_1^y\right)^{\iota' \cdot x + r}}\right)^{1/\iota^* - \iota'} \\
&= \left(\left(g_1^y\right)^{\iota^* \cdot x + r - \iota' \cdot x - r}\right)^{1/\iota^* - \iota'} \\
&= \left(g_1^{y \cdot x\left(\iota^* - \iota'\right)}\right)^{1/\iota^* - \iota'} \\
&= g_1^{y \cdot x}.
\end{aligned} \tag{27}$$

ϵ' is defined as the success probability $\mathrm{ADV}_{\mathrm{CDH}}(.)$ that $\mathscr{F}$ solves the CDH problem. We base this on the forking lemma theorem in [45, 46] and obtain the success probability that $\mathscr{F}$ uses $\mathscr{A}_U$ to solve the CDH problem as follows:

$$\begin{aligned}
\epsilon' &\geq \mathrm{frk} \geq \mathrm{acc}\left(\frac{\mathrm{acc}}{q_H} - \frac{1}{2^l}\right) \\
\mathrm{frk} &\geq \epsilon\left(\frac{\epsilon}{q_H} - \frac{1}{2^l}\right) \\
\mathrm{frk} &\geq \frac{\epsilon^2}{q_H} - \frac{\epsilon}{2^l} \\
\mathrm{frk} &> \frac{\epsilon^2}{q_H} \\
\epsilon' &> \frac{\epsilon^2}{q_H} \\
\therefore \epsilon &< \sqrt{q_H \epsilon'}.
\end{aligned} \tag{28}$$

Note that $acc = \epsilon$, since the simulation behaves naturally, and it does not need to abort the experiment in any event. $\epsilon/2^l$ is omitted since it is negligible. To summarise the probability, $\mathscr{A}_U$ wins the above game and outputs a signcryption $\mathscr{C}^*$ on a message M^* with a probability less than $\sqrt{q_H \epsilon'}$. Therefore, the results of the above probability analysis lead us to conclude that the success of breaking the existential unforgeability of the LBS scheme is nonnegligible due to the probability of breaking the CDH problems that are nonnegligible in the random oracle model. □

4.3. Indistinguishability

Theorem 2. *Against the adaptive chosen ciphertext distinguisher $\mathscr{A}_I$ in the selective-credential exposed attack model, the cyphertext of the level-based signcryption scheme is existentially indistinguishable if CDH and DBDH assumptions hold in the random oracle model.*

Proof. Let us assume that an adversary $\mathscr{A}_I$ exists. It runs the existentially indistinguishable game defined in Section 3.2. Then, it successfully outputs a correct guess. We will demonstrate that we can use an adversary $\mathscr{F}$ to output a solution to the DBDH problem by using $\mathscr{A}_I$ as a tool. Given g, g^x, g^y, g^z, and **Z** as an instance of the DBDH problem, $\mathscr{F}$ runs setup and sets $g = g, o = g^y$ and obtains param = $(p, \widehat{e}, g, o, h)$.

$\mathscr{F}$ assigns $b = z$ and runs TKeyGen to obtain TA's public-private keys. $\mathscr{F}$ also sets $x = x$ and runs SKeyGen to obtain the signer's public key pk_E. Assume that there exists an algorithm managing the list of each query, and such algorithms will be omitted. $\mathscr{F}$ builds the queries in the following functions.

HO oracle: given a string Γ as input, **HO** checks whether or not Γ is in the queried list for a hash value request of $h(\Gamma)$. If it exists in the list, then return the corresponding value; otherwise, **HO** randomly chooses $\iota \xleftarrow{\$} \mathbb{Z}_p$ and then returns $h(\Gamma) = \iota$. Note that **HO** keeps (Γ, ι) in the list and only $\mathscr{F}$ can access the list.

VCO queries: $\mathcal{F}$ randomly chooses an integer $d \xleftarrow{\$} \mathbb{Z}_{n+1}^*$. On input $\mathtt{L} = l$, if $l \geq d$, then output $\perp$. Otherwise, **VCO** randomly chooses the integer $k_c \in \mathbb{Z}_p$, if k_c has yet to be selected. $\mathbb{C}_V$ is computed as follows:

$$\begin{aligned} k_1 \xleftarrow{\$} \mathbb{Z}_p &: k_1 = k_2 + k_c \\ C_0 &= o^{k_2}, \\ C_l &= o^{\left(\left(\gamma_0 \cdot \mu_0 + \gamma_l \cdot \mu_l\right) \cdot a + k_1 - k_2 \cdot \gamma_0\right)/\gamma_l}. \end{aligned} \tag{29}$$

VCO then returns $\mathbb{C}_V = (C_0, C_l)$.

SSO queries: let $\mathtt{L}_{sso}$ be the list that stores generated signcryption. On input $\mathtt{LP} =$ "$\mathtt{L} \geq l$" and a message M, if $l \geq d$, then output $\perp$. Otherwise, **SSO** generates a level-based signcryption ciphertext with the following algorithms.

$$\begin{aligned} &r, \iota \xleftarrow{\$} \mathbb{Z}_p, \\ &R = \hat{e}\left(W_l, o^r\right) \cdot \hat{e}\left(X_0, o^{\mu_0 \cdot a + k_c \cdot \gamma_0^{-1}}\right)^{\iota} \prod_{i=l}^{n} \hat{e}\left(X_i, U_i\right)^{\iota}: \\ &\sigma_1 = h\left(R \| \mathrm{pk}_E \| \mathrm{pk}_{\mathrm{TA}} \| \mathtt{LP} \| \sigma_3\right) = \iota, \\ &\sigma_2 = o^r, \sigma_3 = R \oplus M, \\ &\Gamma_1 = R \| \mathrm{pk}_E \| \mathrm{pk}_{\mathrm{TA}} \| \mathtt{LP} \| \sigma_3. \end{aligned} \tag{30}$$

Note that **SSO** has an access to the list of (Γ, ι) via $\mathcal{F}$. **SSO** uses this advantage to update (Γ, ι) to the list in **HO**. **SSO** then responds with $\mathcal{C} = (\sigma_1, \sigma_2, \sigma_3)$ and keeps $(\mathcal{C}, M)$ in $\mathtt{L}_{sso}$.

Optional privacy: let $\mathbb{X}_R = (X_{R,0} = g_1^{x_R \cdot \gamma_0}, \ldots, X_{R,n} = g_1^{x_R \cdot \gamma_n})$ be the receiver's public key pk_R and $\mathbb{X}_E = (X_{E,0} = X_0, \ldots, X_{E,n} = X_n)$ be the sender's public key pk_E. **SSO** generates a level-based signcryption with the following algorithms.

$$\begin{aligned} &r, \iota, \bar{\iota} \xleftarrow{\$} \mathbb{Z}_p, \\ &R = \hat{e}\left(W_l, o^r\right) \cdot \hat{e}\left(X_{E,0}, o^{\mu_0 \cdot a + k_c \cdot \gamma_0^{-1}}\right)^{\iota} \prod_{i=l}^{n} \hat{e}\left(X_{E,i}, U_i\right)^{\iota}, \\ &\widehat{R} = \hat{e}\left(X_{R,0}, o^{\mu_0 \cdot a + k_c \cdot \gamma_0^{-1}}\right)^{\iota} \prod_{i=l}^{n} \hat{e}\left(X_{R,i}, U_i\right)^{\bar{\iota}}, \\ &h\left(\sigma_3\right) = \bar{\iota}, \\ &\sigma_1 = h\left(\widehat{R} \| \overline{R} \| R \| \mathrm{pk}_E \| \mathrm{pk}_R \| \mathrm{pk}_{\mathrm{TA}} \| \mathtt{LP} \| \sigma_3\right) = \iota, \\ &\sigma_2 = o^r, \sigma_3 = R \oplus M, \\ &\Gamma_1 = \widehat{R} \| \overline{R} \| R \| \mathrm{pk}_E \| \mathrm{pk}_R \| \mathrm{pk}_{\mathrm{TA}} \| \mathtt{LP} \| \sigma_3. \end{aligned} \tag{31}$$

Note that $\overline{R}$ does not need to be computed. It only needs to be verified by the following equation when $\mathcal{A}_U$ submits the query of the hash value for σ_1.

$$\hat{e}\left(\overline{R}, g\right) \stackrel{?}{=} \hat{e}\left(X_{R,0}, X_{E,0}\right)^{\bar{\iota}}. \tag{32}$$

If it matches with $X_{R,0}$ in the list of verifier public key and the list of queried ciphertext, it then returns the corresponding result. Otherwise, it returns the new ι_1.

USO queries: on input $M, \mathcal{C}$, and $\mathtt{LP} =$ "$\mathtt{L} \geq l$", if $\mathcal{C} \in \mathtt{L}_{sso}$, then **USO** responses with M from the corresponding $\mathcal{C}$ in the list. Otherwise, **USO** responses with "reject." Note that this setting is based on the assumption that the unforgeability of LBS holds. Hence, all signcryption ciphertexts that are not generated by **SSO** are all invalid signcryption ciphertexts. If the adversary aborts, due to an unnatural simulation, we can then use this adversary to run the unforgeability simulation to solve the CDH problem.

At the beginning of a game, $\mathcal{A}_I$ is given access to the above oracles. Next, we run the simulation between $\mathcal{A}_I$ and $\mathcal{F}$ as modelled in Section 3.2:

(1) Phase 1: $\mathcal{A}_I$ with adaptive strategy makes queries to **SSO**, **VCO**, and **USO** oracles. The oracles responses are as we previously described.

(2) Challenge: at the end of the first phase, $\mathcal{A}_I$ decides to challenge and outputs M_0, M_1, and $\mathtt{LP}^*$. $\mathcal{F}$ aborts the game if

  (1) On input M_0, M_1, and $\mathtt{LP}^*$, $\mathcal{A}_I$ makes a query for a level-based signcryption ciphertext to **SSO** queries

  (2) $\mathcal{A}_I$ has a credential that is equal or higher than the security level assigned in the level-based security policy $\mathtt{LP}^*$

  Otherwise, $\mathcal{F}$ selects a random bit $b^* \in \{0, 1\}$ and computes a response as follows:

$$\begin{aligned} &r, \iota^* = \xleftarrow{\$} \mathbb{Z}_p, \\ &R = Z^{\iota^*} \cdot \hat{e}\left(W_l, o^r\right) \cdot \hat{e}\left(X_0, o^{\mu_0 \cdot a}\right)^{\iota^*} \prod_{i=l}^{n} \hat{e}\left(X_i, U_i\right)^{\iota^*}: \\ &\sigma_2^* = o^r, \sigma_3^* = R \oplus M_{b^*}, \\ &\sigma_1^* = h\left(R \| \mathrm{pk}_E \| \mathrm{pk}_{\mathrm{TA}} \| \mathtt{LP}^* \| \sigma_3^*\right) = \iota^*, \\ &\Gamma^* = R \| \mathrm{pk}_E \| \mathrm{pk}_{\mathrm{TA}} \| \mathtt{LP}^* \| \sigma_3^*. \end{aligned} \tag{33}$$

  Note that $\mathcal{F}$ has access to the list (Γ^*, ι^*). $\mathcal{F}$ uses this advantage to update (Γ^*, ι^*) to the list in **HO**. $\mathcal{F}$ then responds with $\mathcal{C}^* = (\sigma_1^*, \sigma_2^*, \sigma_3^*)$ to $\mathcal{A}_I$.

  Optional privacy: let $\mathbb{X}_R = (X_{R,0} = g_1^{x_R \cdot \gamma_0}, \ldots, X_{R,n} = g_1^{x_R \cdot \gamma_n})$ be the receiver's public key pk_R and $\mathbb{X}_E = (X_{E,0} = X_0, \ldots X_{E,n} = X_n)$ be the sender's public key pk_E. $\mathcal{F}$ computes the challenge ciphertext as follows:

$$\begin{aligned} &r, \iota^*, \bar{\iota}^* \xleftarrow{\$} \mathbb{Z}_p, \\ &R = Z^{\iota^*} \cdot \hat{e}\left(W_l, o^r\right) \cdot \hat{e}\left(X_{E,0}, o^{\mu_0 \cdot a}\right)^{\iota^*} \prod_{i=l}^{n} \hat{e}\left(X_{E,i}, U_i\right)^{\iota^*}: \\ &\widehat{R} = \hat{e}\left(X_{R,0}, o^{\mu_0 \cdot a + k_c \cdot \gamma_0^{-1}}\right)^{\iota} \prod_{i=l}^{n} \hat{e}\left(X_{R,i}, U_i\right)^{\bar{\iota}^*}, \\ &h\left(\sigma_3^*\right) = \bar{\iota}^*, \\ &\sigma_1^* = h\left(\widehat{R} \| \overline{R} \| R \| \mathrm{pk}_E \| \mathrm{pk}_R \| \mathrm{pk}_{\mathrm{TA}} \| \mathtt{LP} \| \sigma_3\right) = \iota^*, \\ &\sigma_2^* = o^r, \sigma_3^* = R \oplus M_{b^*}, \\ &\Gamma^* = \widehat{R} \| \overline{R} \| R \| \mathrm{pk}_E \| \mathrm{pk}_R \| \mathrm{pk}_{\mathrm{TA}} \| \mathtt{LP} \| \sigma_3^*. \end{aligned} \tag{34}$$

Note that $\overline{R}$ does not need to be computed or it can be obtained from the previous queries by computing $\overline{R} = \overline{R}^{\iota^*/\iota'}$. It only needs to be verified by the following equation when $\mathscr{A}_U$ submits the query of the hash value for σ_1.

$$\widehat{e}(\overline{R}, g) \stackrel{?}{=} \widehat{e}\left(X_{R,0}, X_{E,0}\right)^{\iota^*}. \tag{35}$$

If it matches with $X_{R,0}$ in the list of verifier public key and the list of queried ciphertext, it then returns the corresponding result. Otherwise, it returns the new ι_1.

(3) Phase 2: in this phase, $\mathscr{A}_I$ can go back to Phase 1 as many as it requires. However, $\mathscr{F}$ will abort the game if

(1) On input M_0, M_1, and $\mathtt{LP}^*$, $\mathscr{A}_I$ issues a request for a level-based signcryption ciphertext to **SSO** queries
(2) On input $\mathscr{C}^*$ and $\mathtt{LP}^*$, $\mathscr{A}_C$ issues a request for an unsigncryption query to **USO**
(3) $\mathscr{A}_I$ has a credential that is equal or higher than the security level assigned to the level-based security policy $\mathtt{LP}^*$
(4) Guessing: on the valid challenge $M_0, M_1, \mathtt{LP}^*$, $\mathscr{C}^*$, $\mathscr{A}_I$ finally outputs a guess b'

Let $\mathrm{ADV}_{\mathrm{IND-CCA}} = \epsilon$ be an advantage probability that $\mathscr{A}_I$ wins the challenge in the above simulation. A upper bound of queries in polynomial time that $\mathscr{A}_I$ request a hash value to the **HO** oracle is denoted as q. Note that $q \geq q_H$ and $q \ll p$. Since only $\mathscr{F}$ and **SSO** access **HO** before it outputs a response, we can conclude that $q_H \geq q_S$. Therefore, we can analyze the advantage that $\mathscr{A}_I$'s guess is correct and wins the above game as follows:

(i) E_1: during the request of queries to **VCO**, $\mathscr{F}$ did not abort. Let q_{VC} be the highest security level that $\mathscr{A}_I$ issues to the **VCO** oracle, rather than the number of queries that it makes to the **VCO** oracle. Since $\mathscr{A}_I$ can only make one query for the security level $\mathtt{L} = n - 1$, $\mathscr{A}_I$ can use this credential to verify and decrypt ciphertexts 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 $\mathscr{A}_I$ can make a request for credentials, up to the security level $q_{VC} = n - 1$, to the **VCO** oracle, and the value of d is in range of $\{1, \ldots, n\}$. However, if $q_{VC} \geq d$, then **VCO** will always terminate the experiment. Otherwise, $q_{VC} < d$, and **VCO** will not terminate the experiment. To choose q_{VC} and d randomly, the probability that $\mathscr{A}_I$ chooses q_{VC} is $1/n$ and the probability that $\mathscr{F}$ chooses d is $1/n$ Therefore, the probability of this event is $1/n^2$.

(ii) E_2: after the request of queries to **USO**, $\mathscr{A}_I$ did not abort. Let $\widehat{\epsilon}$ be the success probability of solving the CDH problem. Since the probability of $\mathscr{A}_I$ breaking unforgeability is equal to $\widehat{\epsilon}$, the probability of this event is more than $1 - \sqrt{q_H \widehat{\epsilon}}$.

(iii) E_3: after Phase 1 and Phase 2, $\mathscr{F}$ did not abort. Since we have assumed that $\mathscr{A}_I$ follows the experiment and outputs a guess with a valid challenge $(\mathtt{LP}^*, \mathscr{C}^*)$, the probability of this event is 1.

The advantage that $\mathscr{A}_I$ successfully completes the challenge in the above simulation and generates a right educational guess $b' = b$ is $\Pr[\mathrm{ADV}_{\mathrm{IND-}}\ \mathrm{CCA}] \cdot \Pr\ [\mathrm{ADV}_{\mathrm{IND-CCA}}\ |E_1|E_2|E_3|] \geq \epsilon \cdot (1 - \sqrt{q_H \widehat{\epsilon}})1/n^2$.

Let an advantage probability in solving the DBDH problem be denoted by ϵ'. $\mathscr{F}$ is using $\mathscr{A}_I$'s guess to answer the DBDH problem. Due to the condition that $\mathscr{A}_I$ can choose a challenge level-based security policy $\mathtt{LP}^*$, $\mathscr{A}_I$ cannot have the credential above the security level in the challenge level-based security policy $\mathtt{LP}^*$. Hence, there is an event where $\mathscr{A}_I$'s guess in the game is not the right guess for the DBDH problem, where $\mathtt{LP}^* \neq$ "$\mathtt{L} \geq d$". This event probability is $1/n$. To conclude, the advantage probability of $\mathscr{F}$ using $\mathscr{A}_I$ to produce a correct solution for the DBDH problem is $\epsilon' \geq \epsilon \cdot (1 - \sqrt{q_H \widehat{\epsilon}}) \cdot 1/n^2 \cdot 1/n = \epsilon \cdot 1/n^3$. Against the adaptive chosen message and selective-credential exposure attack model, the advantage probability of $\mathscr{A}_I$ breaking the existential indistinguishability of the LBS scheme is $\epsilon \leq n^3 \epsilon' / (1 - \sqrt{q_H \widehat{\epsilon}})$. Note that $n \ll q_H \ll q$. Therefore, the results of the above probability analysis lead us to conclude that the success of breaking the existential indistinguishability of the LBS scheme is nonnegligible due to the probability of breaking the CDH and DBDH problems is nonnegligible in the random oracle model. □

5. Theoretical Analysis

We introduced the notion of a level-based signcryption (LBS) scheme to capture the need for confidential and authenticated messages sent to a specific group of verifiers that satisfy the required security level. The LBS scheme has a short ciphertext, a constant credential size, and an efficient signcrypting algorithm. The encryption security of our LBS scheme is CCA-secure and relies on the CDH and DBDH assumptions.

The communication and computation cost of our scheme and related schemes are compared, as given in Table 2. l is denoted as a security level which is specified in the level-based security policy (MP = "$\mathtt{L} \geq l$"). The total security levels or the total number of attributes are denoted by n, and let l_n be the number of attribute specified in the policy.

The computation cost for the exponential in groups G_1 or G_T is denoted by E, while the computation cost of the multiplication in groups G_1 or G_T is denoted by M. The computation cost for the bilinear pairing function is denoted as P. Computation for hash functions in Z_p is denoted as H. A computation unit given in Table 2 is equivalent to the XOR operation's computation. For lightweight cryptography, S denotes a unit of time used for computing a symmetric encryption scheme such as AES.

We implemented our scheme based on the pairing-based cryptography (PBC) library. The comparison with the existing efficient schemes in PBC is shown in Figures 2 and 3.

Table 2: Comparison cost with three ABE schemes.

Schemes/cost	Communication (size)	Computation (encryption)	Computation (decryption)
BSW07 [10]	$\lvert\mathbb{G}_T\rvert + (2n+1)\lvert\mathbb{G}_1\rvert$	$h + M + (2n+2)E + P$	$(3n+2)E + (2n-1)M + (2n+1)P$
RW15 [11]	$(n+1)\lvert\mathbb{G}_T\rvert + 3n\lvert\mathbb{G}_1\rvert$	$h + E + M + P$	$nh + nE + (4n)M + 3nP$
YTC14 [12]	$(n+2)\lvert\mathbb{G}_1\rvert$	$h + S + nM$	$h + S + nM$
YLWXY20 [40]	$(n+7)\lvert\mathbb{G}_1\rvert + \lvert p\rvert$	$(2l_n + 2n + 7)E + S, +(n + l_n + 9)M + S$	$(3l_n + 13)E + S + 7P, +(n + 3l_n + 2)M$
Our LBS	$\lvert\mathbb{G}_1\rvert + \lvert p\rvert + \lvert\mathbb{G}_T\rvert$	$H + 2E + P + 1$	$1 + H + ((n-l)+2)P + E + nM$
Our LBS with OP	$\lvert\mathbb{G}_1\rvert + \lvert p\rvert + \lvert\mathbb{G}_T\rvert$	$1 + 2H + (n-l+2)P, +2E + nM$	$1 + 2H + (2n-l+2)P + 2E + 2nM$

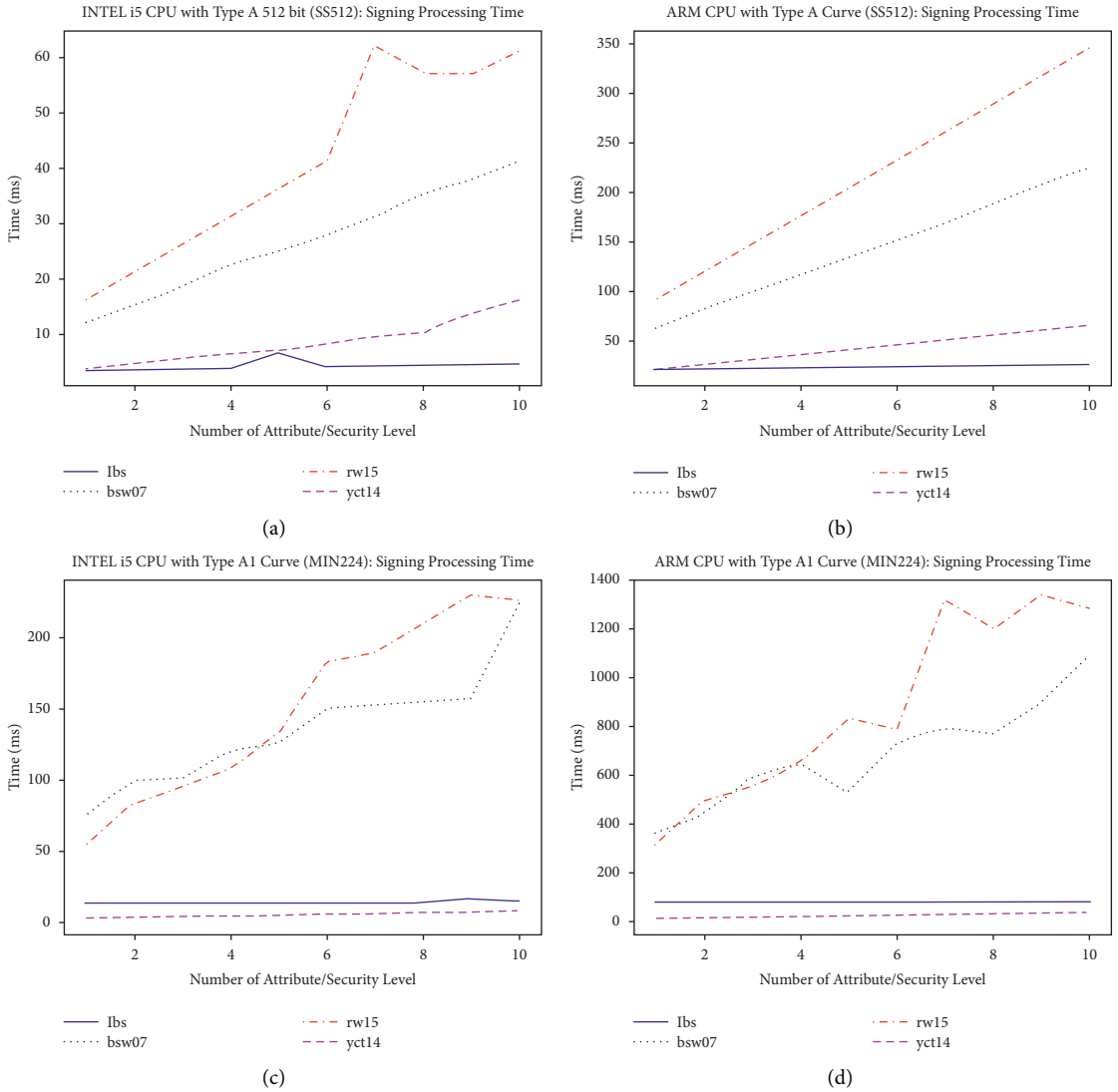

Figure 2: Signcryption ciphertext generation processing time. (a) Type A curve on Intel. (b) Type A curve on ARM. (c) Type A1 curve on Intel. (d) Type A1 curve on ARM.

Noted that, the results of some scheme may not be accurate due the original code not provided in the PBC library, Hence, only selected efficient ABE schemes in Table 2 are implemented. Moreover, from Table 2, our scheme's ciphertext size is constant compared to other schemes, which increase due to the attributes.

The code was written in Python using the charm-crypto framework developed by Akinyele and others [48] for rapid cryptography development. The first experiment was conducted on an Intel 10th Gen Core i5-10400 with 6 cores and 12 threads configuration with 32 gigabytes of DDR4 memory. The operating system used in the experiment was Ubuntu 18.04. The results of this experiment are presented on the left side of each figure.

The second experiment was conducted on a Raspberry Pi 4 Cortex-A72 (ARM v8) 64 bit SoC with a 1.5 GHz CPU clock speed, 4 cores, and 4 gigabytes of DDR4 memory. Raspbian was the operation system used in the second

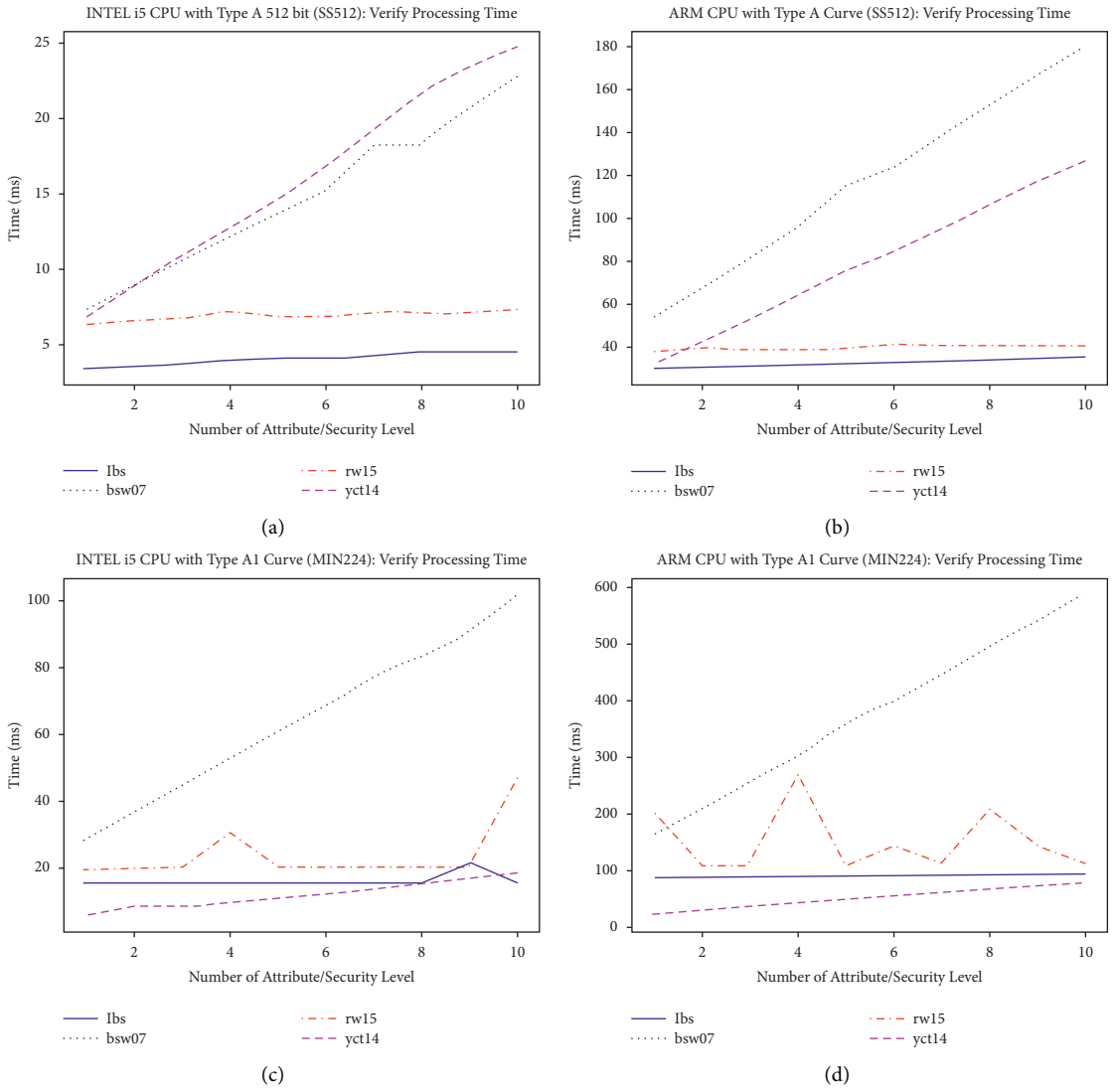

Figure 3: Signcryption verification processing time. (a) Type A curve on Intel. (b) Type A curve on ARM. (c) Type A1 curve on Intel. (d) Type A1 curve on ARM.

Table 3: Curve type A and type A1 parameters used in PBC library [47].

Curve type A	Value
Base field size	512 bits
K	2
DLog security	1024
Q	8780710799663312522437781984754049815806883199414208211028653399266475630880222957078625179422662221423155858769582317459277713367317481324925129998224791
H	1201601226489114607938882136674053420480295440125131182291961513104720728935970453110284480218390653778677 6
R	730750818665451621361119245571504901 405976559617
exp1	107
exp2	159
sign0 and sign1	1
Curve type A1	Value
Base field size	512 bits
K	2
DLog security	1024

Table 3: Continued.

Curve type A	Value
Q	48512875896303752499712277254589628 51641935218829452119818956751100907 31581150453612948393470993158989600 45398524682007334164928531594799149 10054803644576011091315742065569036 18912908584413608071582472594605013 43449199712532828063940008683740048 50098044198971373968965561057845838 81269342426305573976187765392 59
L	1304
n (=r)	36203638728584889925158415861634051 13165623297633919492402206530672318 89239664517621603278709696387305671 98058600508960697138006366861790409 77652838540728366486056523929529131 48442469092845976172822740742242547 33917313218308080644731349763985110 82162719551471174603705642580481969 26320404795750428340438630 89

experiment. The results are presented on the right side of each figure.

The experiments were executed with two different types of curves, namely, type A and type A1. Type A curve is a curve that produces the fastest bilinear pairing computation, and it achieves security comparable to 1024 bits of discrete logarithm (DLog) security. On the other hand, type A1 provides higher security, which is 2048 bits of DLog security. Comparison results using the type A and A1 curves are shown in Figures 2 and 3. Meanwhile, our scheme's performances for the rest of the results are quite good. Only the result where our scheme consumes more time than others is in the setup process. However, this is still acceptable for access control in VANETs.

The parameters of these two curves are given in Table 3. The experiment was conducted 500 times for each scheme to find the average time consumed in each computation process. The message used in the experiments was randomly generated in the G_T domain. From the result in Figures 3 and 2, our scheme shows a significant boost in the encryption and verification in both curve types. Even when compared with a lightweight ABE scheme in [12], our scheme did not have much difference in the performance. Moreover, the size of our ciphertext is constant when the number of attributes increased. Meanwhile, the other schemes are linear to the number of attributes.

6. Conclusion

Privacy issues regarding information shared in organisations without an efficient and proper control mechanism have motivated us to provide a scheme to solve this. The notion of a level-based signcryption scheme provides simplicity, confidentiality, and privacy, enhancing secure communication in VANETs or any ad-hoc network that often needs to broadcast messages. With LBS, the communication among the VANET's nodes, such as RSU nodes and vehicles, is now efficient, secure, and private. LBS ensure the confidentiality, authenticity, and data for RSU to securely broadcast message to nodes. On the other hand, the vehicle node can be confident that its message communicates with RSU that cannot be read by other nodes eventhough it sometimes needs to communicate with RSU via other nodes. Similar to WSN and IoT networks, LBS provides secure broadcast communication among IoT devices. The IoT node can choose to securely and privately communicate with its peer or have private communication with the primary access point units or any devices with a high-level policy setting such as the organisation's server, workstation, or mobile.

Moreover, the proposed scheme is an ideal tool for enabling access control systems or secure shared document systems for a large organisation where a hierarchical structure is applied. A file can be shared or transmitted to the same peer or higher-level security users via a broadcast channel. LBS scheme enables the above scenario securely. Even the message revel to the public, it will not be able to link the signer. In the event of credential disputes or discloses to others, our scheme did not provide a credential revocation mechanism. To resolve this issue, the system should resolve it without reissuing a new credential to other users. Our scheme can use the group key as a key to revoke the credential without affecting honest users. Nevertheless, this is not an ideal solution to this issue. Hence, this will be addressed in our future work.

Data Availability

The source code and some data generated from the experiment are found in the GitHub link (https://github. com/yourchkung/LBSOP).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Y. Maalej, S. Sorour, A. Abdel-Rahim, and M. Guizani, "Vanets meet autonomous vehicles: multimodal surrounding recognition using manifold alignment," *IEEE Access*, vol. 6, pp. 29026–29040, 2018.

[2] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: common practices and emerging technologies," *IEEE Access*, vol. 8, pp. 58443–58469, 2020.

[3] S. Kumar and A. K. Verma, "Position based routing protocols in VANET: a survey," *Wireless Personal Communications*, vol. 83, no. 4, pp. 2747–2772, 2015.

[4] A. Kaur Malhi, S. Batra, and H. Singh Pannu, "Security of vehicular ad-hoc networks: a comprehensive survey," *Computers & Security*, vol. 89, 2020.

[5] R. G. Engoulou, M. Bellaïche, S. Pierre, and A. Quintero, "VANET security surveys," *Computer and Communications*, vol. 44, no. 1–13, 2014.

[6] A. Katiyar, D. Singh, and R. S. Yadav, "State-of-the-art approach to clustering protocols in VANET: a survey," *Wireless Networks*, vol. 26, no. 7, pp. 5307–5336, 2020.

[7] A. Mchergui, T. Moulahi, B. Alaya, and S. Nasri, "A survey and comparative study of qos aware broadcasting techniques in VANET," *Telecommunication Systems*, vol. 66, no. 2, pp. 253–281, 2017.

[8] S. Amit and B. Waters, "Fuzzy identity-based encryption," in *Proceedings of the Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 457–473, Springer, Aarhus, Denmark, May 2005.

[9] V. Goyal, O. Pandey, S. Amit, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006*, pp. 89–98, ACM, Alexandria, VA, USA, October 30 - November 3 2006.

[10] J. Bethencourt, S. Amit, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the 2007 IEEE Symposium on Security and Privacy (S&P 2007), 20-23 May 2007*, pp. 321–334, IEEE Computer Society, Oakland, California, USA, May 2007.

[11] Y. Rouselakis and B. Waters, "Efficient statically-secure large-universe multi-authority attribute-based encryption," in *Rainer Böhme and Tatsuaki Okamoto, editors, Financial Cryptography and Data Security - 19th International Conference, FC 2015*, pp. 315–332, Springer, San Juan, Puerto Rico, January 2015.

[12] X. Yao, Z. Chen, and Y. Tian, "A lightweight attribute-based encryption scheme for the internet of things," *Future Generation Computer Systems*, vol. 49, pp. 104–112, 2015.

[13] K. Xue, J. Hong, Y. Xue, D. S. L. Wei, N. Yu, and P. Hong, "Cabe: a new comparable attribute-based encryption construction with 0-encoding and 1-encoding," *IEEE Transactions on Computers*, vol. 66, no. 9, pp. 1491–1503, 2017.

[14] W. Bagga and R. Molva, "Policy-based cryptography and applications," in *Financial Cryptography and Data Security*, pp. 72–87, Springer, 2005.

[15] M. Bellare and G. Fuchsbauer, "Policy-based signatures," *Public-Key Cryptography - PKC 2014*, Springer, in *Public-Key Cryptography - PKC 2014 - 17th International Conference on Practice and Theory in Public-Key Cryptography*, pp. 520–537, March 2014.

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

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

[18] H. K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-based signatures," in *Topics in Cryptology - CT-RSA 2011 - the Cryptographers' Track at the RSA Conference 2011*, pp. 376–392, Springer, San Francisco, CA, USA, February 2011.

[19] S. Shahandashti and R. Safavi-Naini, "Threshold attribute-based signatures and their application to anonymous credential systems," in *Progress in Cryptology - AFRICACRYPT 2009, Second International Conference on Cryptology*, pp. 198–216, Springer, Africa, Gammarth, Tunisia, June 2009.

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

[21] J. Herranz, F. Laguillaumie, B. Libert, and C. Ràfols, "Short attribute-based signatures for threshold predicates," in *Topics in Cryptology - CT-RSA 2012 - the Cryptographers' Track at the RSA Conference 2012*, pp. 51–67, Springer, San Francisco, CA, USA, February 27 - March 2, 2012.

[22] A. Escala, J. Herranz, and P. Morillo, "Revocable attribute-based signatures with adaptive security in the standard model," in *Progress in Cryptology - AFRICACRYPT 2011 - 4th International Conference on Cryptology in Africa*, pp. 224–241, Springer, Dakar, Senegal, July 2011.

[23] T. Okamoto and K. Takashima, "Decentralized attribute-based signatures," in *Public-Key Cryptography - PKC 2013 - 16th International Conference on Practice and Theory in Public-Key Cryptography*, pp. 125–142, Springer, Nara, Japan, February 26 - March 1, 2013.

[24] 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, 2014.

[25] T. Okamoto and K. Takashima, "Efficient attribute-based signatures for non-monotone predicates in the standard model," in *Proceedings of the Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in Public Key Cryptography*, pp. 35–52, Springer, Taormina, Italy, March 6-9, 2011, Proceedings, volume 6571 of Lecture Notes in Computer Science.

[26] Y. Sakai, N. Attrapadung, and G. Hanaoka, "Attribute-based signatures for circuits from bilinear map," in *Proceedings of the Public-Key Cryptography - PKC 2016 - 19th IACR International Conference on Practice and Theory in Public-Key Cryptography*, pp. 283–300, Springer, Taipei, Taiwan, March 6-9, 2016, Proceedings, Part I, volume 9614 of Lecture Notes in Computer Science.

[27] Li Jin, M. Ho Au, W. Susilo, D. Xie, and K. Ren, "Attribute-based signature and its applications," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, ASIACCS 2010*, pp. 60–69, ACM, Beijing, China, April 13-16, 2010.

[28] L. Zhang, Y. Mu, and Q. Wu, "Compact anonymous hierarchical identity-based encryption with constant size private keys: table 1," *The Computer Journal*, vol. 59, no. 4, pp. 452–461, 2016.

[29] P. Chen, Y. Wu, J. Su, and X. Wang, "Comparing performance of hierarchical identity-based signature schemes," *IEICE - Transactions on Info and Systems*, vol. E99.D, no. 12, pp. 3181–3184, 2016.

[30] D. Boneh, X. Boyen, and E.-J. Goh, "Hierarchical identity based encryption with constant size ciphertext," in *EUROCRYPT, Volume 3494 of Lecture Notes in Computer Science*Springer, Lecture Notes in Computer Science, pp. 440–456, 2005.

[31] Y. Zheng, "Digital signcryption or how to achieve cost(signature & encryption) ≪ cost(signature) + cost(encryption)," *Advances in Cryptology - CRYPTO '97*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 165–179, 1997.

[32] M. Gagné, S. Narayan, and R. Safavi-Naini, "Threshold attribute-based signcryption," in *Security and Cryptography for Networks, 7th International Conference, SCN 2010, amalfi, Italy, september 13-15, 2010*Springer, Lecture Notes in Computer Science. Proceedings, Volume 6280 of Lecture Notes in Computer Science, pp. 154–171, 2010.

[33] T. Pandit, S. K. Pandey, and R. Barua, "Attribute-based signcryption: signer privacy, strong unforgeability and IND-CCA security in adaptive-predicates model (extended version)," *J. Internet Serv. Inf. Secur.*vol. 6, no. 3, pp. 61–113, 2016.

[34] Y. Song, Z. Li, Y. Li, and J. Li, "Attribute-based signcryption scheme based on linear codes," *Information Sciences*, vol. 417, pp. 301–309, 2017.

[35] H. Zheng, J. Qin, J. Hu, and Q. Wu, "Threshold attribute-based signcryption and its application to authenticated key agreement," *Security and Communication Networks*, vol. 9, no. 18, pp. 4914–4923, 2016.

[36] Y. S. Rao, "Attribute-based online/offline signcryption scheme," *International Journal of Communication Systems*, vol. 30, no. 16, Article ID e3322, 2017.

[37] H. Yu and S. Wang, "Certificateless threshold signcryption scheme with secret sharing mechanism," *Knowledge-Based Systems*, vol. 221, Article ID 106981, 2021.

[38] Y. Wang, H. Pang, R. H. Deng, Y. Ding, Q. Wu, and B. Qin, "Securing messaging services through efficient signcryption with designated equality test," *Information Sciences*, vol. 490, pp. 146–165, 2019.

[39] L. Huy Quoc, D. Dung Hoang, and R. Partha Sarathi, "Lattice-based signcryption with equality test in standard model," *Computer Standards and Interfaces*, vol. 76103515 pages, 2021.

[40] J. Yu, S. Liu, S. Wang, Y. Xiao, and B. Yan, "LH-ABSC: a lightweight hybrid attribute-based signcryption scheme for cloud-fog-assisted IoT," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 7949–7966, 2020.

[41] P. Vijayakumar, M. Azees, A. Kannan, and L. Jegatha Deborah, "Dual authentication and key management techniques for secure data transmission in vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1015–1028, 2016.

[42] M. Wazid and S. Mohammad, "Obaidat, ashok kumar das and pandi vijayakumar. "Sac-fiiot: secure access control scheme for fog-based industrial internet of things"" in *Proceedings of the IEEE Global Communications Conference, GLOBECOM 2020, Virtual Event*, pp. 1–6, IEEE, Taiwan, December 7-11, 2020.

[43] X. Xia, S. Ji, P. Vijayakumar, J. Shen, J. Joel, and P. C. Rodrigues, "An efficient anonymous authentication and key agreement scheme with privacy-preserving for smart cities," *International Journal of Distributed Sensor Networks*, vol. 17, no. 6, Article ID 155014772110268, 2021.

[44] B. Bera, A. K. Das, M. S. Obaidat, P. Vijayakumar, K.-F. Hsiao, and Y. Park, "AI-enabled blockchain-based access control for malicious attacks detection and mitigation in IoE," *IEEE Consumer Electronics Magazine*, vol. 10, no. 5, pp. 82–92, 2021.

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

[46] M. Bellare and G. Neven, "Multi-signatures in the plain public-key model and a general forking lemma," in *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006*, pp. 390–399, ACM, Alexandria, VA, USA, October 30 - November 3, 2006.

[47] PBC Library: http://crypto.stanford.edu/pbc..

[48] A. Joseph, G Christina, and M Ian, ""Charm: a framework for rapidly prototyping cryptosystems"," *Journal of Cryptographic Engineering*, vol. 3, no. 2, pp. 111–128, 2013.