IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received February 5, 2020, accepted March 6, 2020, date of publication April 13, 2020, date of current version June 2, 2020.
Digital Object Identifier 10.1109/ACCESS.2020.2987336

Aggregatable Certificateless Designated Verifier Signature

PAIRAT THORNCHAROENSRI, WILLY SUSILO, (Senior Member, IEEE), AND JOONSANG BAEK
Institute of Cybersecurity and Cryptology, University of Wollongong, Wollongong, NSW 2522, Australia
Corresponding author: Pairat Thorncharoensri (pairat@uow.edu.au)

ABSTRACT In recent years, the Internet of Things (IoT) devices have become increasingly deployed in many industries and generated a large amount of data that needs to be processed in a timely and efficient manner. Using aggregate signatures, it provides a secure and efficient way to handle large numbers of digital signatures with the same message. Recently, the privacy issue has been concerned about the topic of data sharing on the cloud. To provide the integrity, authenticity, authority, and privacy on the data sharing in the cloud storage, the notion of an aggregatable certificateless designated verifier signature scheme (ACLDVS) was proposed. ACLDVS also is a perfect tool to enable efficient privacy-preserving authentication systems for IoT and or the vehicular ad hoc networks (VANET). Our concrete scheme was proved to be secured underling of the Computational Diffie-Hellman assumption. Compared to other related schemes, our scheme is efficient, and the signature size is considerably short.

INDEX TERMS Certificateless signature, aggregate signature, designated verifier, privacy, authentication, vehicular ad hoc network (VANET), wireless sensor network (WSN).

I. INTRODUCTION

A wireless sensor network (WSN) is made up of a large number of sensor nodes, which are densely deployed very close to each other. It has the advantages of low cost, high efficiency and low latency. The protocols and algorithms used in the wireless sensor network must possess self-organizing capabilities. A sensor node has an onboard processor, and it can be used to process simple computations locally and transmits only the necessary and partially processed data back to the requested node. This cooperative effort of sensor nodes is one of the unique and attractive features of wireless sensor networks.

The above-described feature ensures a wide range of applications for wireless sensor networks, for example, healthcare, military, and security. For healthcare application, a doctor can securely monitor the wearable health devices. With consent from the patient, the wearable health devices allow the doctor to have a better understanding of the patient's current condition. However, the generated patient's medical reports from these devices could leak the privacy of the patient, and, hence, there should be appropriately handled and be protected. The data and its signature transmitting between the sensor devices and the monitor have to be encrypted to provide confidentiality and authenticity. It can be done merely implementing the SSL/TLS protocol. Nevertheless, it ensures only confidentiality and authenticity, but it does not preserve the privacy of the patient. For the privacy of a patient, it can be achieved by implement the designated verifier signature instead of the general signature. Moreover, to save the cost of the communication and computation, the aggregate signature scheme can be applied to limit the amount of the data needed to transmit through the network. The data and its signature can be gathering at the sensor gateway, then the sensor gateway aggregates them into a single signature and passes it to the monitor server. The detail of this implementation is described in [10].

With the rapid growth of the application of various Internet of Things (IoT) devices and the development of wireless communication technology especially for WSN, the topic of vehicular ad hoc networks (VANET) has attracted significant interest and attention. In VANET, Vehicles equipped wireless devices can communicate with each other. The main objective of VANET is to set up and maintain a communication network among vehicles without using any centralized network architecture based base station [26]. One of the

The associate editor coordinating the review of this manuscript and approving it for publication was Kashif Saleem.

examples of the VANET applications, the critical medical emergencies in a place with no access to any communication infrastructure, it is vital to pass on the information that could save human lives. Lack of support in VANET, it has put additional responsibility on each vehicle that is part of the network. Every node must maintain and forward the communication on this network to other nodes. In the United State, the intelligent transportation systems (ITS) implement the Dedicated Short Range Communications (DSRC) that operates around the 5.9 GHz frequency band. The DSRC consists of RoadSide units (RSUs) and On-Board Units (OBUs) that have transceivers and transponders. A vehicle with OBUs can communicate with another vehicle with OBUs directly, which is called Vehicle to Vehicle (V2V) communication. Meanwhile, a vehicle with OBUs that communicates with a Road Side Unit (RSU) is known as Vehicle-to-Infrastructure (V2I). Each vehicle in VANET can operate in both modes of communication simultaneously. More details of VANET can be found in [25].

However, the rapid movement of the nodes affect the stability of the network route and the large scale of nodes in the network caused communication delays, is a significant problem on VANET that could not be ignored [9]. The concept of certificateless public key cryptography has been recommended to secure the communication in the VANET, and avoid the complexity associated with managing public key certificates and the drawbacks of the key escrows in identity-based cryptography [25], [26], [29], [30]. For the public key management in cryptography, Certificate Authority(CA) is commonly utilized to certify the public key. However, it is a security weak point in the VANET, which creates a single point of failure. In the VANET enviroments, where the perspective of limited bandwidth and the dynamic nature of the networks are crucial. A compromised AC will put the security of the whole VANET in risk, and the collapsing of the communication in the network is unavoidable. Hence, the efficient key agreement and distribution in VANET is strategically assigned to certificateless cryptography. In some specific applications, the signatures on the same message generated by different nodes need to be compressed to reduce the cost of transmission and verification computation due to the bandwidth and storage constrained environments. The above issue can be solved with an aggregate signature which can reduce the cost of verification, and the length of the signature. It was designed to be effective in the bandwidth and storage constrained environments.

A. RELATED WORK

Since the seminal introduction of digital signature notion [11] and its formalization [12], the notion of digital signatures has been extended to capture different scenarios and situations in real life. With a public-private keys pair, it allows a signer with a private key to produce a signature on the message and lets anyone verify this signature with a public key.

A designated verifier signature (DVS) provides both authenticity and deniability properties at the same time. It was proposed by Jakobsson, Sako and Impagliazzo in [19]. The authenticity property ensures that a signer indeed signs this digital signature. The deniability property ensures that only the designated verifier can verify the validity of this digital signature signed by the signer. Moreover, this conviction cannot be transferred to any other third party. It has been widely studied and extended to many areas [13], [16], [18], [21], [22], [24], [32]–[35], [37].

Laguillaumie and Vergnaud were first to propose a multi-designated verifiers signature scheme (MDVS) in [21]. Later, Thorncharoensri *et al.* [35] introduced a policy-controlled signature (PCS) which is a variant of MDVS in the attributed based topics. They also proposed the extension schemes in [37]. The signature size of MDVS schemes is linear to the number of the designated verifiers, while the signature size of PCS schemes is linear to the number of the attribute in the policy, but it does not limit the number of the designated verifiers. There are many variants of DVS, such as the universal designated verifier signature (UDVS) scheme [32], [36] where a delegator can sign on behalf of the signer, the one-time UDVS scheme where a signature can be recover if the delegator produced more than one universal designated verifier signature, ID-based DVS [33], and proxy DVS [16].

The certificateless public key cryptography was first proposed by Al-Riyami and Paterson in [2]. Unlike the traditional public key cryptography that needs a certificate to ensure the authenticity of the public keys, certificateless public key cryptography does not require the use of any certificate. The formal security definitions of certificateless signature (CLS) schemes have been intensively discussed by Au *et al.* in [3], Huang *et al.* in [15] and Huang *et al.* in [17]. Karati *et al.* [20] put forward the lightweight certificateless signature that can be run on restricted computation devices. However, it was proven to be insecure by Zhang *et al.* [39]. Later, Yang *et al.* [38] illustrates the public key replacement attack on the Zhang *et al.*'s improved CLS scheme [39].

The used of certificateless aggregate signature on VANETs application was demonstrated by Cui *et al.* [9]. They also proposed an efficient certificateless aggregate signature scheme for the VANETs which does not require bilinear pairing. The certificateless public key cryptography in the standard model was proposed by Canard and Trinh in [7].

In some applications such as Multicast applications which may allow data sending from the leaf nodes to gather at the branch node before pass to the root node. This leads to a many-to-one communication pattern. To ensure authenticity, integrity and non-repudiation, the cost of verification computation and bandwidth is linear to the size of the leaf nodes. This leads to the propose of the aggregate signature scheme in 2003 by Boneh *et al.* [6]. An aggregate signature refers to an aggregation of n signatures of n messages signed by n signers, by an aggregation algorithm, into a single signature. The verifier only needs to verify this aggregate signature, which confirms whether or not the signature is from the specified n users. Aggregate signatures not only reduce the cost of verification but can also reduce the length of the signature that is transmitted and can be valuable in environments constrained for bandwidth and storage. Since the proposed of the aggregate signature scheme, it has been widely studied and expanded in many areas. Recently, due to the popularity of the IoT topics, the compact and lightweight certificateless aggregate signature schemes were proposed in [10], [14], [23]. Deng *et al.* proposed a certificateless short aggregate signature in [10]. It is efficient in the signing and verifying process where requires only two pairing operations in the verification, and the size of the signature is only one point on the elliptic curve and some state of information. However, the state of information must be shared among signers (devices) before each signer can sign on a message. This causes another issue in securely generated a shared state of information. Hashimoto and Ogata introduced a compact and unrestricted certificateless aggregate signature which the signature's size is constant. Their concrete scheme shares the similarity to Deng *et al.*'s concrete scheme; however, the former scheme is much flexible. It does not need to share a state of information for every time the signer generates a signature. Li *et al.* recently proposed the most efficient certificateless aggregate signature scheme in [23]. The concrete scheme does not require the bilinear pairing, and it also allowed the scalar multiplication over E/F_q to be computed offline and store them for later use. Therefore, it is suitable for limited computation power IoT devices.

Huang *et al.* was first introduced the notion of certificateless designated verifier signature schemes in [18]. Recently, many certificateless designated verifier signature schemes were proposed [13], [22], [28], [31]. Rastegari *et al.* [28] provided intensive security reviews on certificateless designated verifier signature schemes and they gave a conclusion on the suitable security model for certificateless designated verifier signature schemes. They also proposed a concrete scheme in a standard model. Shen *et al.* introduced the certificateless aggregate signature with the designated verifier (CLASDV) in [31]. In this scheme, the aggregator acts as a delegator in the universal designated verifier signature scheme which, given a signature from the original signer, he/she can generate a designated verifier signature on behalf of the original signer. Unlike the aggregator in our scheme, he/she can only aggregate the signature to reduce the communication cost and cannot generate a designated verifier on behalf of the signers.

Our goal to construct the aggregatable certificateless designated verifier signature scheme (ACLDVS) is not a simple task by combining or modifying the above-mentioned works. There is no generic DVS scheme that can convert an existing DVS scheme to ACLDVS. Combining certificateless signature, aggregate signature and designated verifier signature together is not a trivial process. For example, in Shen *et al.* CLASDV scheme [31], it can only achieve privacy through the aggregator. The privacy of the signer is not preserved since the beginning.

Since our scheme is unique and applicable for many applications, our comparison with other schemes is aims to compare in term of performance. our scheme position in the balance of communication cost, performance and privacy-preserving. Hence, the well-known related signature schemes [9], [18], [20] were chosen for the comparison in Section V.

B. OUR CONTRIBUTIONS

In this paper, we concentrate on providing a designated verifier signature that can simultaneously aggregate by any party; however, only the designated verifier can prove the validity of this aggregate designated verifier signature.

Our reliable and efficient certificateless aggregate designated verifier scheme solves the aforementioned problems in integrity, authentication, and privacy. Compared with other certificateless aggregate signature and certificateless designated verifier schemes, our scheme has better performance as follows.

1) Our concrete scheme does not employ expensive bilinear pairings and map-to-point hash functions, hence, our scheme can easily implement on most of IoT devices. Since our scheme has a unique property that it is a combination of aggregate signature, designated verifier signature and certificate less signature schemes, we compared our scheme to well known efficient schemes in those areas. The results of the comparison are in the Table 2 and Figure 1 to 5.
2) Our concrete scheme satisfies the requirements of unforgeability in [28] as we demonstrated our security proofs in Section IV-A.
3) Our concrete scheme provides a signer privacy preservation in the aspect of deniability; hence, none other than designated verifier can verify the validity of the signature. This property is due to the transcript simulation, which it is indicated that the designated verifier can also generate the signature.

Paper Organization: The organization of the paper is organized as follows. In the next section, some notation and definitions used throughout this paper is described. The definition of an aggregatable certificateless designated verifier signature (ACLDVS) and its security notions are described in Section III. In the following section, the construction of the efficient ACLDVS scheme is described with its security proof. Finally, the comparison of our scheme with other schemes and the conclusion of the paper will be presented in the last two sections.

II. PRELIMINARIES

A. NOTATION

The following notations will be used in the rest of this paper. A function $f : \mathbb{N} \rightarrow \mathbb{R}$ is *negligible* when, for all constant $c > 0$ and for all sufficiently large n, $f(n) < \frac{1}{n^c}$. $poly(.)$ is a deterministic polynomial function. Let $[n]$ represent a series of numbers(or indexes), e.g., if n is integer then $[n] = \{0, \ldots, n\}$. Hence, for all polynomials $poly(k)$ and for all sufficiently large k, we say that q is polynomial-time in k if

$q \leq poly(1^k)$. Denote by $l \xleftarrow{\$} L$ the operation of picking l at random from a (finite) set L. Let $H : \{0, 1\}^* \to \mathbb{G}_1$ be a collision-resistant hash function. Let $h : \{0, 1\}^* \to \mathbb{Z}_p^*$ be a collision-resistant hash function. Let e be the base of the Natural Logarithms.

B. BILINEAR PAIRING

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

1) *Bilinearity:* $\forall (g_1 \in \mathbb{G}_1; g_2 \in \mathbb{G}_2; a, b \in \mathbb{Z}_p) : \hat{e}(g_1^a, g_2^b) = \hat{e}(g_1, g_2)^{ab}$.
2) *Non-degeneracy:* $\exists\, g_1 \in \mathbb{G}_1 \,\exists\, g_2 \in \mathbb{G}_2 : \hat{e}(g_1, g_2) \neq 1$.
3) *Computability:* $\exists\, \hat{e} : \forall\, g_1 \in \mathbb{G}_1, \forall\, g_2 \in \mathbb{G}_2; \hat{e}(g_1, g_2) \in \mathbb{G}_T$

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

C. COMPLEXITY ASSUMPTIONS

Definition 1 (Computational Diffie-Hellman (CDH) Problem): Given a 3-tuple $(\mathbf{g}, \mathbf{g}^{\chi}, \mathbf{g}^{\psi} \in \mathbb{G}_1)$ as input, output $\mathbf{g}^{\chi \cdot \psi}$. An algorithm $\mathcal{A}$ has advantage ϵ' in solving the CDH problem if

$$\Pr\left[\mathcal{A}(\mathbf{g}, \mathbf{g}^{\chi}, \mathbf{g}^{\psi}) = \mathbf{g}^{\chi \cdot \psi}\right] \geq \epsilon'$$

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

Assumption 1 (Computational Diffie-Hellman Assumption [5], [11]): We say that the (t, ϵ')-CDH assumption holds if no PPT algorithm with time complexity $t(.)$ has an advantage at least ϵ' in solving the CDH problem.

III. AGGREGATABLE CERTIFICATELESS DESIGNATED VERIFIER SIGNATURE SCHEMES (ACLDVS)

In this section we will propose our aggregatable certificateless designated verifier signature schemes (ACLDVS). There are three main players which are a trusted authority *KGC* who issues keys associated with its public key to the rest,a verifier V and a signer S who generates a signature that can be verified *only* by a specified verifier V. Let $\mathtt{ID} = \{ID_1, \ldots, ID_n\}$ be a set of n identities and $\mathtt{U} = \mathtt{ID} \cup \{pk_i : ID_i \in \mathtt{ID}\}$ be a set of identity and public key of n users.

System Parameter Generation (**Setup**):

Given a security parameter ℓ as input, a probabilistic algorithm **Setup** outputs the system parameter param and the private key (sk_K) of a trusted authority. That is,

$$\textbf{Setup}(1^{\ell}) \to \mathsf{param}, sk_K.$$

Extract Partial Private Key (**PPK**):

Given param, a user identity $ID_U \in \{0, 1\}^*$ and sk_K as input, a probabilistic algorithm **PPK** outputs the partial private key (psk_U) and the public parameter (ppk_U) of a user. That is,

$$\textbf{PPK}(\mathsf{param}, ID_U, sk_K) \to (ppk_U, psk_U).$$

Noted that U is represented the user who may be a signer or a verifier.

Setup User Secret Value (**SetSV**):

Given param and a user identity ID_U as input, a probabilistic algorithm **SetSV** outputs the Secret Value (sv_U) of the user. That is,

$$\textbf{SetSV}(\mathsf{param}, ID) \to sv_U.$$

Setup User Private Key (**SetSK**):

Given param, the user identity IDP, psk_U and sv_U as input, a probabilistic algorithm **SetSK** outputs the private key (sk_U) of the user. That is,

$$\textbf{SetSK}(\mathsf{param}, ID, psk_U, sv_U) \to sk_U.$$

Setup User Public Key (**SetPK**):

Given param and the user identity ID_U as input, a probabilistic algorithm **SetPK** outputs the public key (pk_U) of a signer. That is,

$$\textbf{SetPK}(\mathsf{param}, ID_U) \to (pk_U).$$

Signature Signing (**Sign**):

Given param, sk_S, pk_S, pk_V and a message M as input, a probabilistic algorithm **Sign** outputs a signer's signature δ. That is,

$$\textbf{Sign}(\mathsf{param}, M, sk_S, pk_S, pk_V) \to \delta.$$

Aggregate (**Aggregate**):

Given param, $\mathtt{U}$, $\delta_1, \ldots, \delta_n$ and M as input, a probabilistic algorithm **Aggregate** outputs a signer's signature σ That is,

$$\textbf{Aggregate}(\mathsf{param}, \mathtt{U}, M, \delta_1, \ldots, \delta_n) \to \sigma.$$

Verification (**Verify**):

Given param, sk_V, $\mathtt{U}$, M and σ as input, a deterministic algorithm **Verify** outputs a verification decision $\mathsf{d} \in \{\mathtt{accept}, \mathtt{reject}\}$. That is,

$$\textbf{Verify}(\mathsf{param}, M, \sigma, \mathtt{U}, sk_V) \to \mathsf{d}.$$

Transcript Simulation (**Sim**):

Given param, sk_V, $\mathtt{U}$, M as input, a probabilistic algorithm **Sim** outputs a simulated signature σ That is,

$$\textbf{Sim}(\mathsf{param}, M, \mathtt{U}, sk_V) \to \hat{\sigma}.$$

A. SECURITY MODEL OF AGGREGATABLE CERTIFICATELESS DESIGNATED VERIFIER SIGNATURE SCHEMES

In ACLDVS, there are two models of the attack which describe with different capabilities:

Type I: In this type of the attack, an adversary $\mathcal{A}_I$ does not have access to the master key. However, it has an

TABLE 1. Queries.

Queries	meaning
$\mathbf{Hash-Q}$	a hash query
$\mathbf{PPK-Q}$	a partial private key query $\mathbf{PPK-Q}: ID_i \rightarrow (ppk_i, psk_i)$
$\mathbf{SV-Q}$	a secret key query $\mathbf{SV-Q}: ID_i \rightarrow sv_i$
$\mathbf{SK-Q}$	a secret key query $\mathbf{SK-Q}: ID_i \rightarrow sk_i$
$\mathbf{PK-Q}$	a public key query $\mathbf{PK-Q}: ID_i \rightarrow pk_i$
$\mathbf{RPK-Q}$	a public key replacing (updating) query that on given input $(ID_i, \hat{pk_i})$ and it replaces pk_i with $\hat{pk_i}$.
$\mathbf{Sign-Q}$	a signature query $\mathbf{Sign-Q}$: $(ID_S, ID_V, M) \rightarrow \delta$
$\mathbf{Verify-Q}$	a signature verification query $\mathbf{Verify-Q}$: $(M, \mathtt{U}, \sigma) \rightarrow \{0,1\}$

ability to replace any public key and/or obtain the most of signers' secret keys (but at least one signer's secret key and one verifier's secret key must remain secret to $\mathcal{A}_I$). Given the above ability with the public parameter and queries in Table 1, $\mathcal{A}_I$ can choose messages with adaptive strategies and submit them to the signing oracle. Finally, if $\mathcal{A}_I$ can output a valid message-signature pair that have never been queried before, then $\mathcal{A}_I$ is successful in the attack.

Type II: In this type of the attack, an adversary $\mathcal{A}_{II}$ has access to the master key. However, it doesn't have an ability to replace any public key of its own choice. Given the above ability with the public parameter and queries in Table 1, $\mathcal{A}_I$ can choose messages with adaptive strategies and submit them to the signing oracle. Finally, if $\mathcal{A}_I$ can output a valid message-signature pair that have never been queried before, then $\mathcal{A}_I$ is successful in the attack.

IV. OUR SCHEME

The ACLDVS scheme is described as follows.

Setup : On input a security parameter ℓ, *KGC* randomly chooses a prime $\mathrm{p} \approx poly(1^\ell)$. Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ denote three groups of prime order p. Let $\hat{e}$ be the bilinear mapping function, which maps $\mathbb{G}_1$ and $\mathbb{G}_2$ to $\mathbb{G}_T$. The above mapping function is defined as $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. To generate a public parameter, first, select a random integer $a \in \mathbb{Z}_\mathrm{p}^*$. Choose a random generator $g \in \mathbb{G}_1$ and a bilinear mapping function $\hat{e}$. Construct a function $\varphi : \mathbb{G}_1 \rightarrow \mathbb{G}_2$ and compute $o = \varphi(g) \in \mathbb{G}_2$. Select hash functions $H : \{0,1\}^* \rightarrow \mathbb{G}_1$ and $h : \{0,1\}^* \rightarrow \mathbb{Z}_\mathrm{p}^*$. Compute $W = g^a$. Set $sk_K = a$ and $\mathsf{param} = (\mathrm{p}, \hat{e}, g, o, W, H, h)$. Then, **Setup** returns (param, sk_K).

PPK : With param, sk_K and $ID_U \in \{0,1\}^*$ as input, **PPK** randomly generates psk_U as follows: select random an integer $\mu_U \in \mathbb{Z}_\mathrm{p}^*$. Let $ppk_U = (T_U = g^{\mu_U})$ and compute $l_U = h(ID_U||T_U)$. Let $psk_U = d_U = \mu_U + l_U \cdot a$. Then, **PPK** returns (ppk_U, psk_U).

SetSV : With param and $ID_U \in \{0,1\}^*$ as input, **SetSV** randomly generates sv_U as follows: select random integers $x_U, y_U, z_U \in \mathbb{Z}_\mathrm{p}^*$ and set $sv_U = (x_U, y_U, z_U)$.

SetSK : With param, sv_U, psk_U and $ID_U \in \{0,1\}^*$ as input, **SetSK** sets $sk_U = (x_U, y_U, z_U, d_U)$.

SetPK : With param, ppk_U, sv_U and $ID_U \in \{0,1\}^*$ as input, **SetPK** randomly selects an integer $\eta \in \mathbb{Z}_\mathrm{p}^*$. Let $X_U = g^{x_U}, Y_U = g^{y_U}, Z_U = g^{z_U}$. Next, it computes $B_U = g^{\eta_U}, \gamma_U = h(ID_U||B_U||T_U||X_U||Y_U||Z_U), c_U = \eta_U + d_U \cdot \gamma_U$. Finally, it outputs $pk_U = (X_U, Y_U, Z_U, T_U, B_U, c_U)$ for a user U.

Sign : With param, $sk_S, pk_V, ID_S \in \{0,1\}^*, ID_V \in \{0,1\}^*$ and $M \in \{0,1\}^*$ as input, **Sign** randomly generates a signature δ on message M as follows:

1) Select random integers $r_S \in \mathbb{Z}_\mathrm{p}^*$.
2) Verify pk_V by computing $\gamma_V = h(ID_V||B_V||T_V||X_V||Y_V||Z_V)$ and checking whether $g^{c_V} \stackrel{?}{=} B_V \cdot (T_V \cdot W^{l_V})^{\gamma_V}$
3) Compute $l_V = h(ID_V||T_V)$.
4) Compute $R_S = g^{r_S}$; $\beta_S = h(m||R_S||{Y_V}^{z_S}||pk_V||pk_S)$.
5) $\hat{R}_S = (T_V \cdot W^{l_V})^{r_S} \cdot {Z_V}^{\beta_S \cdot d_S}$.
6) Compute $\alpha_S = h(m||\hat{R}_S||R_S||{X_V}^{y_S}||pk_V||pk_S)$.
7) $\bar{\delta}_S = {Y_V}^{r_S} \cdot \hat{R}_S \cdot {X_V}^{\alpha_S(x_S+z_S)}$.
8) $\delta_S = (\bar{\delta}_S, R_S)$.

Aggregate : Let $\mathtt{R} = \{R_1, \ldots, R_n\}$. With param, $\mathtt{U}, \delta_1, \ldots, \delta_n$ and $M \in \{0,1\}^*$ as input, **Aggregate** generates a signature σ on message M as follows:

$$\sigma = (\bar{\sigma} = \prod_{i=1}^{n} \bar{\delta}_i, \mathtt{R})$$

Verify : With param, $\mathtt{U}$, σ, sk_V and $M \in \{0,1\}^*$ as input, **Verify** checks whether the below equations hold or not.

1) Compute $l_i = h(ID_i||T_i) : \forall ID_i \in \mathtt{U}$,
2) Verify each pk_i by computing $\gamma_i = h(ID_i||B_i||T_i||X_i||Y_i||Z_i)$ and checking whether $g^{c_i} \stackrel{?}{=} B_i \cdot (T_i \cdot W^{l_i})^{\gamma_i}$
3) $\beta_i = h(m||R_i||{Z_i}^{y_V}||pk_V||pk_i) : \forall R_i \in \mathtt{R}$,
4) $\hat{R}_i = {R_i}^{d_V} \cdot (T_i \cdot W^{l_i})^{\beta_i \cdot z_V} : \forall R_i \in \mathtt{R}$, and
5) $\alpha_i = h(m||\hat{R}_i||R_i||{Y_i}^{x_V}||pk_V||pk_i) : \forall ID_i \in \mathtt{U}$.
6) Check $\bar{\sigma} \stackrel{?}{=} \prod_{i=1}^{n} {R_i}^{y_V} \cdot \prod_{i=1}^{n} \hat{R}_i \cdot \prod_{i=1}^{n} (X_i \cdot Z_i)^{\alpha_i \cdot x_V}$.

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

Sim : With param, $\mathtt{U}$, $sk_V, ID_V \in \{0,1\}^*$ and $M \in \{0,1\}^*$ as input, **Sim** randomly generates a signature σ on message M as follows:

1) Select random integers $r_1, \ldots, r_n \in \mathbb{Z}_\mathrm{p}^*$.
2) Compute $l_i = h(ID_i||T_V) : \forall ID_i \in \mathtt{U}$.

3) Verify each pk_i by computing $\gamma_i = h(ID_i||B_i||T_i||X_i||Y_i||Z_i)$ and checking whether $g^{c_i} \stackrel{?}{=} B_i \cdot (T_i \cdot W^{l_i})^{\gamma_i}$.
4) Compute $R_i = g^{r_i} : \forall ID_i \in \mathtt{U}$,
5) $\beta_i = h(m||R_i||Z_i^{y_V}||pk_V||pk_i) : \forall ID_i \in \mathtt{U}$,
6) $\hat{R}_i = (R_i)^{d_V} \cdot (T_V \cdot W^{l_V})^{\beta_i \cdot z_V} : \forall ID_i \in \mathtt{U}$, and
7) $\alpha_i = h(m||\hat{R}_i||R_i||Y_i^{x_V}||pk_V||pk_i) : \forall ID_i \in \mathtt{U}$.
8) Finally, compute $\bar{\sigma} = \prod_{i=1}^{n} R_i^{y_V} \cdot \prod_{i=1}^{n} \hat{R}_i \cdot \prod_{i=1}^{n}(X_i \cdot Z_i)^{\alpha_i \cdot x_V}$.
9) Output a signature $\sigma = (\bar{\sigma}, \mathtt{R})$.

A. SECURITY ANALYSIS

Theorem 1: The ACLDVS scheme is designated verifier signature scheme.

Proof: The verification of ACLDVS requires x_U, y_U, z_U and d_U which are the secret key of the designated verifier. From the **Sim** algorithm, the designated verifier also can generate a valid signature by using his/her secret key (x_U, y_U, z_U, d_U). Hence, the signature produced by the designated verifier is ***indistinguishable*** from the signature produced by the signer. To be precise, this signature cannot confirm its validity by a third party. □

Theorem 2: The ACLDVS scheme is existentially unforgeable under Type-I adversary $\mathcal{A}_I$ attack model, if the CDH assumption holds in the random oracle model.

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

Initialization: on input $\mathbf{g}$, $\mathbf{g}^{\chi}$ and $\mathbf{g}^{\psi}$ as an instance of the CDH problem, $\mathcal{F}$ runs **Setup** and sets $g = \mathbf{g}$, $W = g^{\chi}$ and obtains ($\mathsf{param} = (\mathrm{p}, \hat{e}, g, o, W, h)$, $sk_K = \chi$).

Queries: The following queries are constructed by $\mathcal{A}_I$ before running the simulation.

Hash − Q : On a request for a hash value of a string Γ ($h(\Gamma)$), **Hash − Q** check whether Γ in the queried list or not. If it exists in the list then return the corresponding value, otherwise, **Hash − Q** randomly chooses $\iota \xleftarrow{\$} \mathbb{Z}_p$ then returns $h(\Gamma) = \iota$. **Hash − Q** keeps (Γ, ι) in its list and the list can be accessed only by $\mathcal{F}$. Let ϱ_H be a number of the hash queries.

PPK − Q : With $ID_i \in \{0, 1\}^*$ as input, **PPK − Q** randomly generates psk_i as follows: select random integers $\mu_i l_i \in \mathbb{Z}_{\mathrm{p}}^*$. Let $ppk_i = (T_i = g^{\mu_i - l_i \chi})$ and set $h(ID_i||T_i) = l_i$. **PPK − Q** outputs $psk_i = d_i = \mu_i$. However, if the input identity is ID_V^*, will abort the simulation. Let ϱ_{pp} be the list of queried ID_i in the **PPK − Q** queries. **PPK − Q** returns (ppk_U, psk_U).

SV − Q : With $ID_i \in \{0, 1\}^*$ as input, **SV − Q** selects random integers $x_i, y_i, z_i \in \mathbb{Z}_{\mathrm{p}}^*$, sets $sv_i = (x_i, y_i, z_i)$. Then, **SK − Q** returns sv_i for ID_i. However, if the input identity is ID_V^*, it will abort the simulation. Let ϱ_{sk} be a number of the queries with the ID_i in the list $\mathcal{L}_{sk}$.

SK − Q : Since **PPK − Q** and **SV − Q** queries can be used to obtain the same result with this type of query, hence, for this attack model, we simply ignore its construction.

PK − Q : With $ID_i \in \{0, 1\}^*$ as input, **PK − Q** runs **SetPK** to output $pk_i = (X_i = g^{x_i}, Y_i = o^{y_i}, Z_i = o^{z_i}, T_i)$ for a user U. However, if the input identity is ID_S^*, it randomly picks $x_{S^*}, y_{S^*}, \mu_{S^*}, k_{S^*}, j_{S^*}, \in \mathbb{Z}_{\mathrm{p}}^*$, sets $\gamma_{S^*} = k_{S^*} = h(ID_S^*||B_{S^*}||T_{S^*}||X_{S^*}||Y_{S^*}||Z_{S^*})$ in **Hash − Q** and calculates $B_{S^*} = g^{j_{S^*}}(T_{S^*} \cdot W^{l_{S^*}})^{-k_{S^*}}$, $c_{S^*} = j_{S^*}$. It outputs $pk_{S^*} = (X_{S^*} = g^{x_{S^*}}, Y_{S^*} = g^{y_{S^*}}, T_{S^*} = g^{\mu_{S^*}}, \gamma_{S^*}, c_{S^*})$ for ID_S^*. If the input identity is ID_V^*, it randomly selects $x_{V^*}, y_{V^*}, \mu_{V^*}, k_{V^*}, j_{V^*} \in \mathbb{Z}_{\mathrm{p}}^*$, sets $\gamma_{V^*} = k_{V^*} = h(ID_V^*||B_{V^*}||T_{V^*}||X_{V^*}||Y_{V^*}||Z_{V^*})$ in **Hash − Q** and calculates $B_{V^*} = g^{j_{V^*}}(T_{V^*} \cdot W^{l_{V^*}})^{-k_{V^*}}$, $c_{V^*} = j_{V^*}$. It outputs $pk_{V^*} = (X_{V^*} = g^{x_{V^*}}, Y_{V^*} = g^{\mu_{V^*}}, Z_{V^*} = g^{\psi \cdot z_{V^*}}, T_{V^*} = g^{\mu_{V^*}})$ for ID_V^*. Let ϱ_{pk} be a number of the queries with the ID_i in the list $\mathcal{L}_{pk}$.

RPK − Q : With $ID_i \in \{0, 1\}^*$ and $\hat{pk_i}$ as input, **RPK − Q** uses ID_i to get the corresponding pk_i from **SetPK** and replaces it with $\hat{pk_i}$. In order for $\hat{pk_i}$ to pass the public key verification, **RPK − Q** recomputes T_U, B_U, c_U for $\hat{pk_i}$. However, for ID_S^*, **RPK − Q** randomly chooses $k_i, j_i \in \mathbb{Z}_{\mathrm{p}}^*$, sets $\gamma_i = k_i = h(ID_i||B_i||T_i||X_i||Y_i||Z_i)$ and calculates $B_i = g^{j_i}(T_{V^*} \cdot W^{l_{V^*}})^{-k_i}$, $c_i = j_i$. The list $\mathcal{L}_{rpk}$ keeps (ID_i, pk_i,$\hat{pk_i}$). If sv_i that is corresponded with $\hat{pk_i}$ had been queried, then the record (ID_i, pk_i,$\hat{pk_i}$) will be removed from the list. Intuitively, $\mathcal{L}_{rpk}$ keeps only the records that $\mathcal{F}$ cannot simulate the signature due to the lack of knowledge of the corresponding sv. Moreover, if the input identity is ID_V^*, it will abort the simulation. Let ϱ_{rpk} be a number of the queries in the list $\mathcal{L}_{rpk}$.

Sign − Q : With $M \in \{0, 1\}^*$, ID_V and $\mathtt{U}$ as input, **Sign − Q** runs **Sign** or **Sim** for all signature queries except a signature query for ID_S^* as a signer and ID_V^* as a verifier. **Sign − Q** outputs the signature for ID_S^* as a signer and ID_V^* as follows:

1) Select random integers $\dot{r}_{S^*} \in \mathbb{Z}_{\mathrm{p}}^*$.
2) Verify pk_i by computing $\gamma_i = h(ID_i||B_i||T_i||X_i||Y_i||Z_i)$ and checking whether $g^{c_i} \stackrel{?}{=} B_i \cdot (T_i \cdot W^{l_i})^{\gamma_i}$
3) Run **Hash − Q** to obtain $l_{V^*} = h(ID_{V^*}||T_{V^*})$ and $l_{S^*} = h(ID_{S^*}||T_{S^*})$.
4) Let $r_{S^*} = \dot{r}_{S^*} - \frac{\psi \cdot z_{V^*} \cdot \beta_{S^*} \cdot l_{S^*}}{l_{V^*}}$
5) $R_{S^*} = g^{r_{S^*}}$
6) $\beta_{S^*} = h(m||R_{S^*}||Y_{V^*}{}^{z_{S^*}}||pk_{V^*}||pk_{S^*})$
7) $\hat{R}_{S^*} = (T_{V^*} \cdot W^{l_{V^*}})^{r_{S^*}} \cdot Z_{V^*}{}^{\beta_{S^*} \cdot d_{S^*}}$

$$= (g^{\mu_{V^*} + l_{V^*} \cdot \chi})^{(\dot{r}_{S^*} - \frac{\psi \cdot z_{V^*} \cdot \beta_{S^*} \cdot l_{S^*}}{l_{V^*}})} \cdot (g^{\psi \cdot z_{V^*}})^{(\beta_{S^*} \cdot \mu_{S^*} + \beta_{S^*} \cdot l_{S^*} \cdot \chi)}$$

$$= g^{\mu_{V^*} \cdot \dot{r}_{S^*} + l_{V^*} \cdot \chi \cdot \dot{r}_{S^*}} \cdot g^{-\frac{\mu_{V^*} \cdot \psi \cdot z_{V^*} \cdot \beta_{S^*} \cdot l_{S^*}}{l_{V^*}}} \cdot g^{-\frac{l_{V^*} \cdot \chi \cdot \psi \cdot z_{V^*} \cdot \beta_{S^*} \cdot l_{S^*}}{l_{V^*}}}$$

$$
\begin{aligned}
&\cdot g^{\psi\cdot z_{V*}\cdot\beta_{S*}\cdot\mu_{S*}+\chi\cdot\psi\cdot z_{V*}\cdot\beta_{S*}\cdot l_{S*}}\\
&= g^{\mu_{V*}\cdot\dot{r}_{S*}+l_{V*}\cdot\chi\cdot\dot{r}_{S*}}\\
&\cdot g^{-\frac{\mu_{V*}\cdot\psi\cdot z_{V*}\cdot\beta_{S*}\cdot l_{S*}}{l_{V*}}}\\
&\cdot g^{-\chi\cdot\psi\cdot z_{V*}\cdot\beta_{S*}\cdot l_{S*}}\cdot g^{\psi\cdot z_{V*}\cdot\beta_{S*}\cdot\mu_{S*}}\\
&\cdot g^{\chi\cdot\psi\cdot z_{V*}\cdot\beta_{S*}\cdot l_{S*}}\\
&= g^{\mu_{V*}\cdot\dot{r}_{S*}+l_{V*}\cdot\chi\cdot\dot{r}_{S*}}\cdot g^{-\frac{\mu_{V*}\cdot\psi\cdot z_{V*}\cdot\beta_{S*}\cdot l_{S*}}{l_{V*}}}\\
&\cdot g^{\psi\cdot z_{V*}\cdot\beta_{S*}\cdot\mu_{S*}}\\
&= g^{\mu_{V*}\cdot\dot{r}_{S*}}\cdot(\mathbf{g}^{\chi})^{l_{V*}\cdot\dot{r}_{S*}}\\
&(\mathbf{g}^{\psi})^{z_{V*}\cdot\beta_{S*}(\mu_{S*}-\frac{\mu_{V*}\cdot l_{S*}}{l_{V*}})}.
\end{aligned}
$$

8) Obtain $\alpha_{S*} = h(m||\hat{R}_{S*}||R_{S*}||{X_{V*}}^{y_{S*}}||pk_{V*}||pk_{S*})$.
9) $\bar{\delta}_{S*} = {Y_{V*}}^{r_{S*}}\cdot\hat{R}_{S*}\cdot X_{V*}^{\alpha_{S*}(x_{S*}+z_{S*})}$.
10) $\delta_{S*} = (\bar{\delta}_{S*}, R_{S*})$.

Sign − Q outputs δ_{S*} for the query of a signature on M, ID_S^* and ID_V^*. Let ϱ_s be a number of the queries in the list $\mathcal{L}_s$.

Verify − Q : With $M \in \{0,1\}^*$, δ, ID_V and $\mathtt{U}$ as input, **Verify − Q** runs **Verify** for all ID_i. If ID_V is $ID_V{}^*$ and $ID_S{}^* \in \mathtt{U}$, checks whether the below equations hold or not.

1) Run **Hash − Q** to obtain $l_{V*} = h(ID_V^*||T_{V*})$ and $l_i = h(ID_i||T_i) : \forall ID_i \in \mathtt{U}$.
2) $\beta_i = h(m||R_i||{Y_{V*}}^{z_i}||pk_{V*}||pk_i) : \forall R_i \in \mathtt{R}$
3) Compute $\hat{R}_i = {R_i}^{d_{V*}}\cdot(X_{V*})^{d_i} : \forall R_i \in \mathtt{R}$.
4) Make a query for $\alpha_i = h(m||\hat{R}_i||R_i||X_{V*}^{y_i}||pk_V{}^*||pk_i) : \forall ID_i \in \mathtt{U}$ to **Hash − Q**.
5) Check $\hat{e}(\bar{\sigma}, o) \stackrel{?}{=} \prod_{i=1}^{n}\hat{e}(R_i, Y_V)\cdot \hat{e}(\prod_{i=1}^{n}\hat{R}_i, o)\cdot\prod_{i=1}^{n}\hat{e}(X_i\cdot Z_i, X_{V*})^{\alpha_i}$.

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

Phase I: The simulation is begun by giving an access to the above queries to $\mathcal{A}_I$. Noted that $\mathcal{A}_I$ always makes a query for a any string (or message) to **Hash − Q** oracle before it outputs a potential forgery.

Phase II: At the end of the simulation, after executing an adaptive strategy with the above queries, $\mathcal{A}_I$ outputs a forgery σ^* on a message M^* with $\mathtt{ID}_1,\ldots,\mathtt{ID}_n \in \mathtt{U}^* : \exists\mathtt{ID}_i \notin (\mathcal{L}_{sk}\wedge\mathcal{L}_{pp})$ and $\mathtt{ID}_{V*} \notin (\mathcal{L}_{sk}\wedge\mathcal{L}_{pp})$. $\mathcal{A}_I$ wins the game if a signature σ^* on the message M^* with $\mathtt{U}$, $\mathtt{ID}_{V*}$, pk_{V*} is valid and it was not an output from the **Sign − Q** queries.

Solve CDH Problem: To solve CHD problem, the Forking technique in [4], [27] is applied. $\mathcal{F}$ first obtains a signature σ^* on message M^* where $h(m||R_{S*}||{Y_{V*}}^{z_{S*}}||pk_{V*}||pk_{S*}) = \dot{r_S}$ Simultaneously $\mathcal{F}$ resets $\mathcal{A}_I$ to the initial state and repeats again the above simulation with a different hash value $h(m||R_{S*}||{Y_{V*}}^{z_{S*}}||pk_{V*}||pk_{S*}) = \ddot{r_S}$. Eventually, $\mathcal{A}_I$ outputs another signature σ'. Finally, $\mathcal{F}$ computes

$$
\begin{aligned}
\mathbf{Z} &= (\frac{\bar{\sigma}^*}{\bar{\sigma}'})^{\frac{1}{z_{V*}\cdot l_{S*}(\dot{r_S}-\ddot{r_S})}}\\
&= (\frac{\bar{\delta}_S^*}{\bar{\delta}_S'})^{\frac{1}{z_{V*}\cdot l_{S*}(\dot{r_S}-\ddot{r_S})}}\\
&= (\frac{{Z_{V*}}^{\dot{r_S}\cdot d_{S*}}}{{Z_{V*}}^{\ddot{r_S}\cdot d_{S*}}})^{\frac{1}{z_{V*}\cdot l_{S*}(\dot{r_S}-\ddot{r_S})}}\\
&= ((g^{\psi\cdot z_{V*}})^{(\mu_{S*}+l_{S*}\cdot\chi)(\dot{r_S}-\ddot{r_S})})^{\frac{1}{z_{V*}\cdot l_{S*}(\dot{r_S}-\ddot{r_S})}}\\
&= g^{\frac{\psi\cdot\mu_{S*}}{l_{S*}}}g^{\chi\cdot\psi}
\end{aligned}
$$

$$
\mathbf{g}^{\chi\cdot\psi} = (\frac{\mathbf{Z}}{\mathbf{g}^{\psi\frac{\mu_{S*}}{l_{S*}}}})
$$

Probability: Let ϵ_A be the success probability $\mathsf{ADV}_{\mathcal{A}_I}(.)$ that $\mathcal{A}_I$ outputs a forgery. Let ϵ_F be the success probability $\mathsf{ADV}_{EUF-CMA}(.)$ that $\mathcal{A}_I$ wins the above simulation and ϵ_C the success probability $\mathsf{ADV}_{CDH}(.)$ that $\mathcal{F}$ solves the CDH problem. The success probability in solving CDH problem by using $\mathcal{A}_I$ is based on the Forking Lemma in [4], [27]. Some notation will be defined first.

- ϵ_A : The success probability $\mathsf{ADV}_{\mathcal{A}_I}(.)$ that $\mathcal{A}_I$ outputs a forgery.
- ϵ_F: The success probability $\mathsf{ADV}_{EUF-CMA}(.)$ that $\mathcal{A}_I$ wins the above simulation
- ϵ_C; The success probability $\mathsf{ADV}_{CDH}(.)$ that $\mathcal{F}$ solves the CDH problem.
- $\mathcal{E}_1$: The simulation does not abort in **PPK − Q** queries
- $\mathcal{E}_2$: The simulation does not abort in **SV − Q** queries
- $\mathcal{E}_3$: The simulation does not abort in **RPK − Q** queries
- $\mathcal{E}_4$: The simulation does not abort after $\mathcal{A}_I$ outputs the forgery

Noted that it is a fact that $\varrho_H \gg \varrho_s \geq \varrho_{rpk} \approx \varrho_{pp} \approx \varrho_{sv} \approx \varrho_{pk}$ from the nature of the aforementioned simulation. The success probability in solving CDH problem is described as follows:

$$
\begin{aligned}
\epsilon_F &= \epsilon_A\cdot\Pr[\mathcal{E}_1|\mathcal{E}_2|\mathcal{E}_3|\mathcal{E}_4]\\
&= \epsilon_A\cdot(1-\frac{1}{\varrho_{pp}+1})^{\varrho_{pp}}\cdot(1-\frac{1}{\varrho_{sv}+1})^{\varrho_{sv}}\\
&\quad\cdot(1-\frac{1}{\varrho_{rpk}+1})^{\varrho_{rpk}}\cdot\frac{1}{\varrho_{pk}}\\
&= \epsilon_A\cdot\frac{\varrho_{pp}}{e\cdot(\varrho_{pp}+1)}\cdot\frac{\varrho_{sv}}{e\cdot(\varrho_{sv}+1)}\\
&\quad\cdot\frac{\varrho_{rpk}}{e\cdot(\varrho_{rpk}+1)}\cdot\frac{1}{\varrho_{pk}}\\
&\geq \epsilon_A\cdot\frac{\varrho_H}{e\cdot(\varrho_H+1)}\cdot\frac{\varrho_H}{e\cdot(\varrho_H-1)}\\
&\quad\cdot\frac{\varrho_H}{e\cdot(\varrho_H+1)}\cdot\frac{1}{\varrho_{pk}}\\
&\geq \frac{\epsilon_A\cdot{\varrho_H}^3}{e^3\cdot\varrho_{pk}\cdot(\varrho_H+1)^3}\\
\therefore\epsilon_F &\approx \frac{\epsilon_A}{e^3\cdot\varrho_{pk}} \qquad (1)
\end{aligned}
$$

$$
\begin{aligned}
\epsilon_C &\geq \text{frk} \geq \text{acc}(\frac{\text{acc}}{\varrho_H} - \frac{1}{2^l}) \\
\text{frk} &\geq \epsilon_F(\frac{\epsilon_F}{\varrho_H} - \frac{1}{2^l}) \\
\text{frk} &\geq \frac{{\epsilon_F}^2}{\varrho_H} - \frac{\epsilon_F}{2^l} \\
\text{frk} &> \frac{{\epsilon_F}^2}{\varrho_H} \\
\epsilon_C &> \frac{{\epsilon_F}^2}{\varrho_H} \\
\therefore \epsilon_F &< \sqrt{\varrho_H \epsilon_C}
\end{aligned} \tag{2}
$$

From (1) and (2),

$$
\frac{\epsilon_A}{e \cdot {\varrho_{pk}}^2} \leq \sqrt{\varrho_H \epsilon_C}
$$

$$
\therefore \epsilon_A \approx e \cdot {\varrho_{pk}}^2 \cdot \sqrt{\varrho_H \epsilon_C} \tag{3}
$$

Noted that $\frac{\epsilon_F}{2^l}$ is negligible, hence, it is omitted. To summarize the probability, $\mathcal{A}_I$ wins the above game and outputs a signature σ^* on a message M^* with a probability of $e^3 \cdot \varrho_{pk} \cdot \sqrt{\varrho_H \epsilon_C}$. The above success probability shows that our aggregatable certificateless designated verifier signature scheme secures against existentially unforgeable under an adaptive chosen message attack in the Type-I adversary model if the success probability of solving CDH problem is negligible. $\square$

Theorem 3: The ACLDVS scheme is existentially unforgeable under Type-II adversary $\mathcal{A}_I$ attack model, if the CDH assumption holds in the random oracle model.

Proof: For The type-I adversary attack model, hence, the *KGC* secret key is compromised, hence,

Assume that there exists a forger algorithm $\mathcal{A}_{II}$ running the existential unforgeability game defined in Section III-A. Then we will show that, by using $\mathcal{A}_{II}$, an adversary $\mathcal{F}$ solves the CDH problem.

Initialization: on input $\mathbf{g}$, $\mathbf{g}^\chi$ and $\mathbf{g}^\psi$ as an instance of the CDH problem, $\mathcal{F}$ runs **Setup** and sets $g = \mathbf{g}$ and obtains (param $= (\mathrm{p}, \hat{e}, g, o, W, h)$, $sk_K = a$).

Queries: The following queries are constructed by $\mathcal{A}_{II}$ before running the simulation.

Hash − Q : On a request for a hash value of a string Γ ($h(\Gamma)$), **Hash − Q** check whether Γ in the queried list or not. If it exists in the list then return the corresponding value, otherwise, **Hash − Q** randomly chooses $\iota \xleftarrow{\$} \mathbb{Z}_p$ then returns $h(\Gamma) = \iota$. **Hash − Q** keeps (Γ, ι) in its list and the list can be accessed only by $\mathcal{F}$. Let ϱ_H be a number of the hash queries.

PPK − Q : With $ID_i \in \{0, 1\}^*$ as input, **PPK − Q** randomly generates psk_i as follows: select random integers $\mu_i \in \mathbb{Z}_{\mathrm{p}}^*$. Let $ppk_i = (T_i = g^{\mu_i})$ and compute $l_i = h(ID_i||T_i)$. **PPK − Q** outputs $psk_i = d_i = \mu_i + l_i \cdot a$. Let ϱ_{pp} be the list of queried ID_i in the **PPK − Q** queries. **PPK − Q** returns (ppk_i, psk_i).

SV − Q : With $ID_i \in \{0, 1\}^*$ as input, **SV − Q** selects random integers $x_i, y_i, z_i \in \mathbb{Z}_{\mathrm{p}}^*$, sets $sv_i = (x_i, y_i, z_i)$. Then, **SK − Q** returns sv_i for ID_i. However, if the input identity is ID_S^* or ID_V^*, it will abort the simulation. Let ϱ_{sk} be a number of the queries with the ID_i in the list $\mathcal{L}_{sk}$.

SK − Q : With $ID_i \in \{0, 1\}^*$ as input, **SK − Q** selects random integers $x_i, y_i \in \mathbb{Z}_{\mathrm{p}}^*$, runs **PPK − Q** to obtain (ppk_i, psk_i) and **SV − Q** to obtain (x_i, y_i, z_i). Next, it sets $sk_i = (x_i, y_i, z_i, d_i)$. Then, **SK − Q** returns sk_i for ID_i. However, if the input identity is ID_S^* or ID_V^*, it aborts the simulation. Let ϱ_{sk} be a number of the queries with the ID_i in the list $\mathcal{L}_{sk}$.

PK − Q : With $ID_i \in \{0, 1\}^*$ as input, **PK − Q** runs **SetPK** and outputs $pk_i = (X_i = g^{x_i}, Y_i = o^{y_i}, Z_i = o^{z_i}, T_i)$ for a user U. However, if the input identity is ID_S^* or ID_V^*, it randomly picks $x_{S^*}, y_{S^*}, z_{S^*}, \mu_{S^*}, x_{V^*}, y_{V^*}, z_{V^*}, \mu_{V^*}, \eta_{S^*}, \eta_{V^*} \in \mathbb{Z}_{\mathrm{p}}^*$. Next, it computes $B_{S^*} = g^{\eta_{S^*}}$, $\gamma_{S^*} = h(ID_S^*||B_{S^*}||T_{S^*}||X_{S^*}||Y_{S^*}||Z_{S^*})$, $c_{S^*} = \eta_{S^*} + d_{S^*} \cdot \gamma_{S^*}$, $B_{V^*} = g^{\eta_{V^*}}$, $\gamma_{V^*} = h(ID_V^*||B_{V^*}||T_{V^*}||X_{V^*}||Y_{V^*}||Z_{V^*})$, $c_{V^*} = \eta_{V^*} + d_{V^*} \cdot \gamma_{V^*}$. Finally, it outputs pk_{S^*} = $(X_{S^*} = \mathbf{g}^{\chi x_{S^*}}, Y_{S^*} = g^{y_{S^*}}, Z_{S^*} = g^{z_{S^*}}, T_{S^*} = g^{\mu_{S^*}}, B_{S^*}, c_{S^*})$ for a user S^* and pk_{V^*} = $(X_{V^*} = \mathbf{g}^{\psi x_{V^*}}, Y_{V^*} = \mathbf{g}^{\psi y_{V^*}}, Z_{V^*} = g^{z_{V^*}}, T_{V^*} = g^{\mu_{V^*}}, B_{V^*}, c_{V^*})$ for a user V^*. Let ϱ_{pk} be a number of the queries with the ID_i in the list $\mathcal{L}_{pk}$.

Sign − Q : With $M \in \{0, 1\}^*$, ID_V and U as input, **Sign − Q** runs **Sign** or **Sim** for all signature queries except a signature query for ID_S^* as a signer and ID_V^* as a verifier. It will compute the signature for ID_S^* as a signer and ID_V^* as follows:

1) Select random integers $r_{S^*}, \hat{r}_{S^*} \in \mathbb{Z}_{\mathrm{p}}^*$.
2) Run **Hash − Q** to obtain

$$
l_{V^*} = h(ID_{V^*}||T_{V^*}).
$$

3) Verify pk_i by computing

$$
\gamma_i = h(ID_i||B_i||T_i||X_i||Y_i||Z_i)
$$

and checking whether $g^{c_i} \stackrel{?}{=} B_i \cdot (T_i \cdot W^{l_i})^{\gamma_i}$

4) Let $\dot{r}_{S^*} = -(\frac{\chi \cdot x_V{}^* \cdot r_S{}^* \cdot x_S{}^* + \hat{r_S}{}^*}{y_V{}^*})$.
5) $R_{S^*} = g^{\dot{r}_{S^*}}$.
6) $\beta_{S^*} = h(m||R_{S^*}||{Y_{V^*}}^{z_{S^*}}||pk_{V^*}||pk_{S^*})$
7) $\hat{R}_{S^*} = {R_{S^*}}^{d_{V^*}} \cdot {Z_{V^*}}^{\beta_{S^*} \cdot d_{S^*}}$.
8) Set $h(m||\hat{R}_{S^*}||R_{S^*}||{X_{V^*}}^{y_{S^*}}||pk_{V^*}||pk_{S^*}) = r_{S^*}$.
9) $\bar{\delta}_S^* = {Y_{V^*}}^{r_S} \cdot \hat{R}_{S^*} \cdot X_{V^*}^{r_S{}^*(\chi \cdot x_S{}^* + z_S{}^*)}$.
$= g^{-\psi \cdot y_V{}^* (\frac{\chi \cdot x_V{}^* \cdot r_S{}^* \cdot x_S{}^* + \hat{r_S}{}^*}{y_V{}^*})} \cdot \hat{R}_{S^*} \cdot g^{\psi \cdot x_V{}^* \cdot r_S{}^* (\chi \cdot x_S{}^* + z_S{}^*)}$.
$= g^{-\chi \cdot \psi \cdot x_V{}^* \cdot r_S{}^* \cdot x_S{}^* - \psi \cdot \hat{r_S}{}^*)} \cdot \hat{R}_{S^*} \cdot g^{\psi \cdot \chi \cdot x_V{}^* \cdot r_S{}^* \cdot x_S{}^* + \psi \cdot x_V{}^* \cdot r_S{}^* \cdot z_S{}^*)}$.
$= \hat{R}_{S^*} \cdot \mathbf{g}^{\psi (x_V{}^* \cdot r_S{}^* \cdot z_S{}^* - \hat{r_S}{}^*)}$.
10) $\delta_S^* = (\bar{\delta}_S^*, R_S{}^*)$.

Sign − Q outputs δ_S^* for the query of a signature on M, ID_S^* and ID_V^*. Let ϱ_s be a number of the queries in the list $\mathcal{L}_s$.

Verify − Q : With $M \in \{0,1\}^*$, δ, ID_V and $\mathtt{U}$ as input, **Verify − Q** runs **Verify** for all ID_i. If ID_V is $ID_V{}^*$ and $ID_S{}^* \in \mathtt{U}$, checks whether the below equations hold or not.

1) Run **Hash − Q** to obtain $l_{V^*} = h(ID^*_V||T_{V^*})$ and $l_i = h(ID_i||T_i) : \forall ID_i \in \mathtt{U}$.
2) $\beta_i = h(m||R_i||Z_i{}^{z_{V^*}}||pk_{V^*}||pk_i) : \forall R_i \in \mathtt{R}$
3) Compute $\hat{R}_i = R_i{}^{d_{V^*}} \cdot Z_{V^*}{}^{\beta_i \cdot d_i} : \forall R_i \in \mathtt{R}$.
4) Make a query for
$$\alpha_i = h(m||\hat{R}_i||R_i||Y_i^{x_{V^*}}||pk_{V^*}||pk_i) : \forall ID_i \in \mathtt{U}$$
to **Hash − Q**.
5) Check $\hat{e}(\bar{\sigma}, o) \stackrel{?}{=} \prod_{i=1}^{n} \hat{e}(R_i, Y_{V^*}) \cdot \hat{e}(\prod_{i=1}^{n} \hat{R}_i, o) \cdot \prod_{i=1}^{n} \hat{e}(X_i \cdot Z_i, X_{V^*})^{\alpha_i}$.

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

Phase I: The simulation is begun by giving sk_K and an access to the above queries to $\mathcal{A}_{II}$. Noted that $\mathcal{A}_{II}$ always makes a query for a any string (or message) to **Hash − Q** oracle before it outputs a potential forgery.

Phase II: At the end of the simulation, after executing an adaptive strategy with the above queries, $\mathcal{A}_{II}$ outputs a forgery σ^* on a message M^* with $\mathtt{ID}_1, \ldots, \mathtt{ID}_n \in \mathtt{U}^* : \exists \mathtt{ID}_i \notin (\mathcal{L}_{sk} \wedge \mathcal{L}_{pp})$ and $\mathtt{ID}_{V^*} \notin (\mathcal{L}_{sk} \wedge \mathcal{L}_{pp})$. $\mathcal{A}_{II}$ wins the game if a signature σ^* on the message M^* with $\mathtt{U}, \mathtt{ID}_{V^*}, pk_{V^*}$ is valid and it was not an output from the **Sign − Q** queries.

Solve CDH Problem: To solve CHD problem, the Forking technique in [4], [27] is applied. $\mathcal{F}$ first obtains a signature σ^* on message M^* where $h(m||\hat{R}^*_S||R_S{}^*||X_{V^*}^{y_S{}^*}||pk_V{}^*||pk_S{}^*) = \dot{r_S}$ Simultaneously $\mathcal{F}$ resets $\mathcal{A}_{II}$ to the initial state and repeats again the above simulation with a different hash value $h(m||\hat{R}^*_S||R_S{}^*||X_{V^*}^{y_S{}^*}||pk_V{}^*||pk_S{}^*) = \ddot{r_S}$. Eventually, $\mathcal{A}_{II}$ outputs another signature σ'. Finally, $\mathcal{F}$ compute

$$\begin{aligned}
\mathbf{Z} &= (\frac{\bar{\sigma}^*}{\bar{\sigma}'})^{\frac{1}{y_{S^*} \cdot x_{V^*}(\dot{r_S} - \ddot{r_S})}} \\
&= (\frac{\bar{\delta}^*_S}{\bar{\delta}'_S})^{\frac{1}{y_{S^*} \cdot x_{V^*}(\dot{r_S} - \ddot{r_S})}} \\
&= (\frac{(X_{S^*} \cdot Y_{S^*})^{\dot{r_S} \cdot x_{V^*} \cdot \psi}}{(X_{S^*} \cdot Y_{S^*})^{\ddot{r_S} \cdot x_{V^*} \cdot \psi}})^{\frac{1}{y_{S^*} \cdot x_{V^*}(\dot{r_S} - \ddot{r_S})}} \\
&= (g^{\chi \cdot x_{S^*}} \cdot g^{y_{S^*}})^{\frac{\psi \cdot x_{V^*}(\dot{r_S} - \ddot{r_S})}{y_{S^*} \cdot x_{V^*}(\dot{r_S} - \ddot{r_S})}} \\
&= g^{\chi \cdot \psi \cdot \frac{x_{S^*}}{y_{S^*}}} \cdot g^{\psi}
\end{aligned}$$

$$\begin{aligned}
\mathbf{g}^{\chi \cdot \psi} &= (\frac{\mathbf{Z}}{\mathbf{g}^{\psi}})^{\frac{y_{S^*}}{x_{S^*}}} \\
&= (\frac{(\mathbf{g}^{\chi \cdot \psi \cdot \frac{x_{S^*}}{y_{S^*}}} \cdot \mathbf{g}^{\psi})}{\mathbf{g}^{\psi}})^{\frac{y_{S^*}}{x_{S^*}}} \\
&= \mathbf{g}^{\chi \cdot \psi}
\end{aligned}$$

Probability: Let ϵ_A be the success probability $\mathsf{ADV}_{\mathcal{A}_{II}}(.)$ that $\mathcal{A}_{II}$ outputs a forgery. Let ϵ_F be the success probability $\mathsf{ADV}_{EUF-CMA}(.)$ that $\mathcal{A}_{II}$ wins the above simulation and ϵ_C the success probability $\mathsf{ADV}_{CDH}(.)$ that $\mathcal{F}$ solves the CDH problem. The success probability in solving CDH problem by using $\mathcal{A}_{II}$ is based on the Forking Lemma in [4], [27]. Some notation will be defined first.

- ϵ_A : The success probability $\mathsf{ADV}_{\mathcal{A}_{II}}(.)$ that $\mathcal{A}_{II}$ outputs a forgery.
- ϵ_F: The success probability $\mathsf{ADV}_{EUF-CMA}(.)$ that $\mathcal{A}_{II}$ wins the above simulation
- ϵ_C; The success probability $\mathsf{ADV}_{CDH}(.)$ that $\mathcal{F}$ solves the CDH problem.
- $\mathcal{E}_1$: The simulation does not abort in **SK − Q** queries
- $\mathcal{E}_2$: The simulation does not abort after $\mathcal{A}_{II}$ outputs the forgery

Noted that it is a fact that $\varrho_H \gg \varrho_{pk} > \varrho_{sk}$ from the nature of the aforementioned simulation. The success probability in solving CDH problem is described as follows:

$$\begin{aligned}
\epsilon_F &= \epsilon_A \cdot \Pr[\mathcal{E}_1|\mathcal{E}_2] \\
&= \epsilon_A \cdot (1 - \frac{1}{\varrho_{sk} + 2})^{\varrho_{sk}} \cdot \frac{1}{\varrho_{pk} \cdot (\varrho_{pk} - 1)} \\
&= \frac{\epsilon_A (\varrho_{sk} + 1)^2}{e \cdot \varrho_{pk} \cdot (\varrho_{pk} - 1) \cdot (\varrho_{sk} + 2)^2} \\
&\approx \frac{\epsilon_A}{e \cdot \varrho_{pk}{}^2} \\
\therefore \epsilon_F &\approx \frac{\epsilon_A}{e \cdot \varrho_{pk}{}^2} \qquad (4)
\end{aligned}$$

$$\begin{aligned}
\epsilon_C &\geq \text{frk} \geq \text{acc}(\frac{\text{acc}}{\varrho_H} - \frac{1}{2^l}) \\
\text{frk} &\geq \epsilon_F(\frac{\epsilon_F}{\varrho_H} - \frac{1}{2^l}) \\
\text{frk} &\geq \frac{\epsilon_F{}^2}{\varrho_H} - \frac{\epsilon_F}{2^l} \\
\text{frk} &> \frac{\epsilon_F{}^2}{\varrho_H} \\
\epsilon_C &> \frac{\epsilon_F{}^2}{\varrho_H} \\
\therefore \epsilon_F &< \sqrt{\varrho_H \epsilon_C} \qquad (5)
\end{aligned}$$

From (1) and (2),

$$\begin{aligned}
\frac{\epsilon_A}{e \cdot \varrho_{pk}{}^2} &\leq \sqrt{\varrho_H \epsilon_C} \\
\therefore \epsilon_A &\approx e \cdot \varrho_{pk}{}^2 \cdot \sqrt{\varrho_H \epsilon_C} \qquad (6)
\end{aligned}$$

Noted that $\frac{\epsilon_F}{2^l}$ is negligible, hence, it is omitted. To summarize the probability, $\mathcal{A}_{II}$ wins the above game and outputs a signature σ^* on a message M^* with a probability of $e \cdot \varrho_{pk}{}^2 \cdot \sqrt{\varrho_H \epsilon_C}$. The above success probability shows that our aggregatable certificateless designated verifier signature scheme secures against existentially unforgeable under an adaptive chosen message attack in the Type-II adversary model if the success probability of solving CDH problem is negligible. □

TABLE 2. The comparison of three certificateless signature schemes.

Version / Size&Comp.	ACLDVS	CLAS [9]	CLDVS [18]	CLS [20]
PK_K	$\lvert\mathbb{G}_1\rvert$	$2\lvert\mathbb{G}_1\rvert$	$\lvert\mathbb{G}_1\rvert$	$\lvert\mathbb{G}_1\rvert+\lvert\mathbb{G}_T\rvert$
SK_K	$\lvert\mathbb{Z}_p\rvert$	$2\lvert\mathbb{Z}_p\rvert$	$\lvert\mathbb{Z}_p\rvert$	$\lvert\mathbb{Z}_p\rvert$
PPK	$\lvert\mathbb{G}_1\rvert+\lvert\mathbb{Z}_p\rvert$	$3\lvert\mathbb{G}_1\rvert+3\lvert\mathbb{Z}_p\rvert$	$\lvert\mathbb{G}_1\rvert$	$\lvert\mathbb{G}_1\rvert+\lvert\mathbb{Z}_p\rvert$
PK	$5\lvert\mathbb{G}_1\rvert+\lvert\mathbb{Z}_p\rvert$	$4\lvert\mathbb{G}_1\rvert+3\lvert\mathbb{Z}_p\rvert$	$2\lvert\mathbb{G}_1\rvert$	$2\lvert\mathbb{G}_1\rvert$
SK	$\lvert\mathbb{G}_1\rvert+4\lvert\mathbb{Z}_p\rvert$	$2\lvert\mathbb{Z}_p\rvert$	$\lvert\mathbb{G}_1\rvert$	$\lvert\mathbb{G}_1\rvert+2\lvert\mathbb{Z}_p\rvert$
Signature Size	$2\lvert\mathbb{G}_1\rvert$	$\lvert\mathbb{G}_1\rvert+2\lvert\mathbb{Z}_p\rvert$	$\lvert\mathbb{Z}_p\rvert$	$2\lvert\mathbb{G}_1\rvert$
Sign Comp.	$4h+11E+4M$	$h+E$	$h+E+M+P$	$h+2M+2E$
Verify Comp.	$3h+12E+6M$	$2h+3E+2M$	$h+E+M+3P$	$h+M+3E+P$

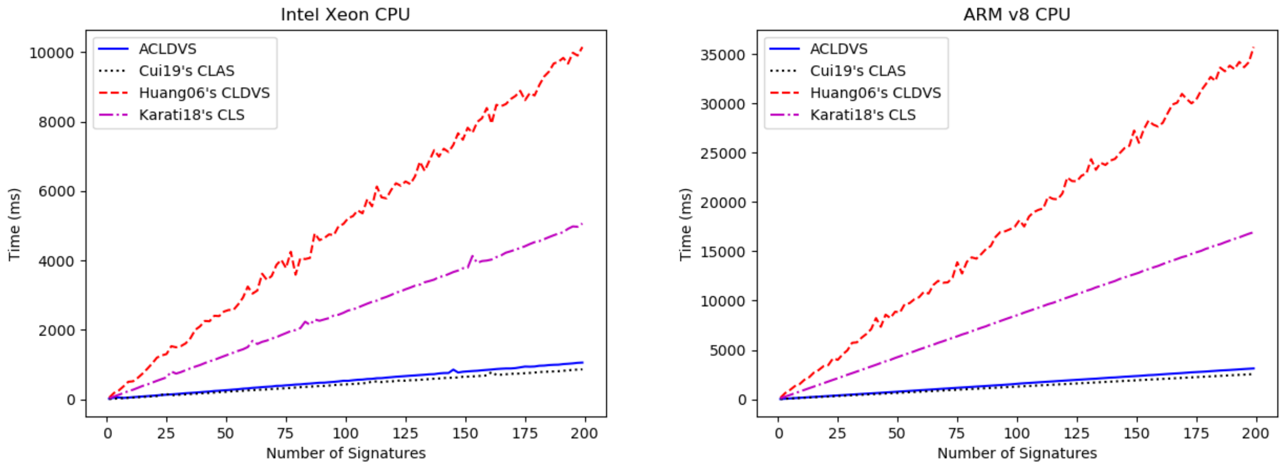

FIGURE 1. Key generation processing time.

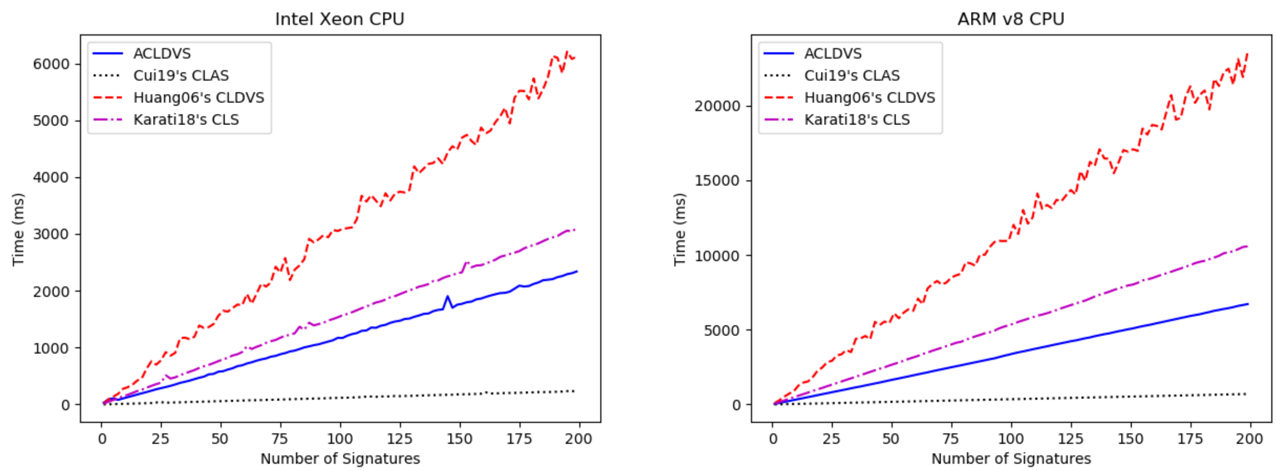

FIGURE 2. Signature generation processing time.

V. ASYMPTOTIC ANALYSIS AND EXPERIMENTAL RESULTS

Our ACLDVS schemes captures the need of authenticity and privacy-preserving in the limited computation environment. The comparison between our scheme and other schemes in Table 2. We denoted n as the number of the signers participated in the signing process for the aggregate signature scheme. Let E denote a computation of exponential in G_1 or G_T. Let M be a computation of scalar multiplication in G_1. Let P be a computation of bilinear pairing function $\hat{e}$. A computation of hash functions from $\{0, 1\}^*$ to G_1 is denoted as H. and the computation of hash function from $\{0, 1\}^*$ to $\mathbb{Z}_p$ is denoted as h. Since the multiplication and addition computation in $\mathbb{Z}_p$ is trivial, they are omitted.

The experiments were using the Pairing-Based Cryptography Library (PBC) provided by [8]. The code was written in Python using the Charm-Crypto framework developed by Akinyele *et al.* [1] for the rapid cryptography

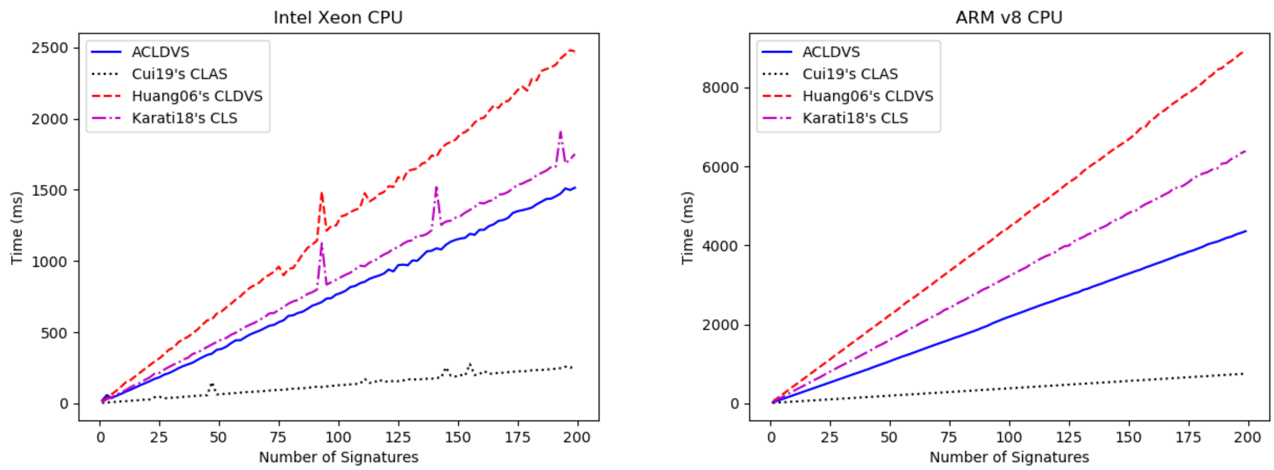

FIGURE 3. Signature verification processing time.

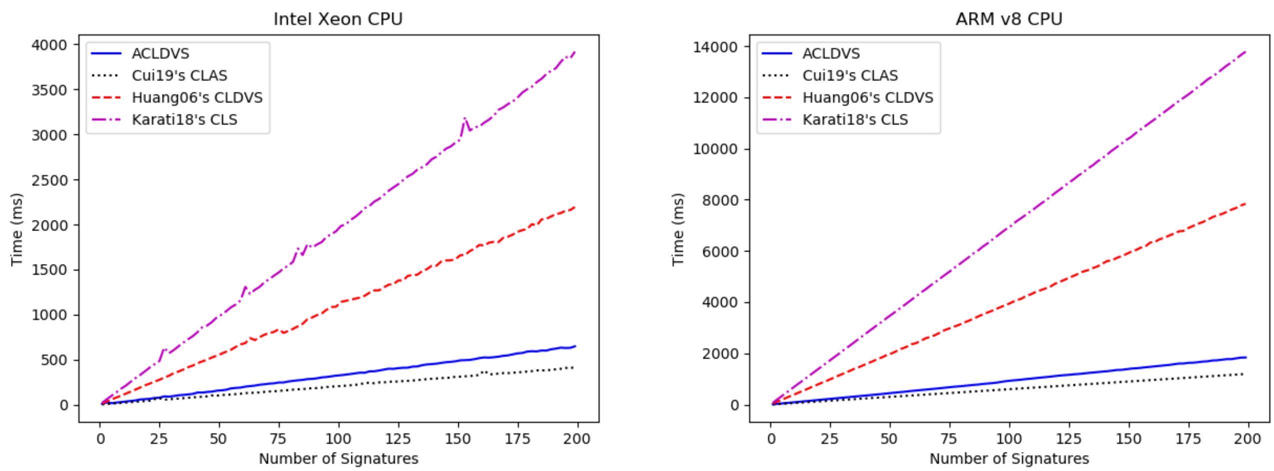

FIGURE 4. Key validation processing time.

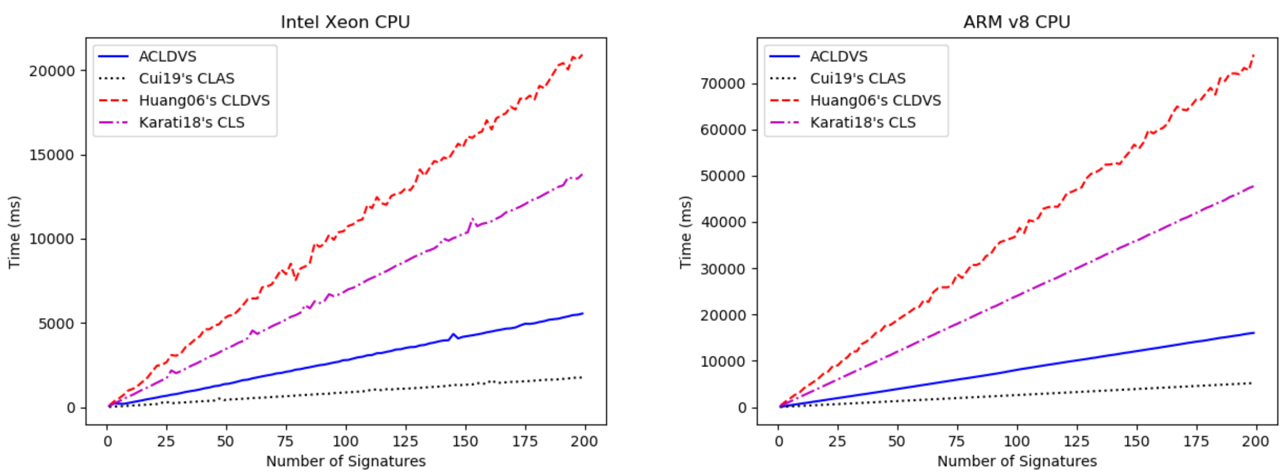

FIGURE 5. Total processing time.

development. The first experiment was conducted on Intel Xeon CPU model X5650 with CPU clocked at 2.67 GHz with 2 cores and 4 threads configuration with 16 Gigabytes of ECC DDR3 memory. The operating system used in this experiment is Ubuntu 18.04. The second experiment was conducted on Raspberry Pi4 Cortex-A72 (ARM v8) 64-bit

SoC with CPU clocked at 1.5 GHz with 4 cores configuration and 4 Gigabytes of DDR4 memory. Raspbian is the operation system used in the second experiment.

both experiments were executed with 224 bit of MNT (Type D in PBC) curves. Type D curve with 224 bit size of group element is a curve that has a short size for the group elements, and it is considerably fast for the bilinear pairing computation. It achieved the security comparable to the 1344 bits (6 x 244 bits) of discrete logarithm (DLog) security.

Each experiment was conducted by first randomly selected one verifier (only for the designated verifier scheme). The number of signers participated in the simulation were start from 1 to 200 signers with a unique identity for each signer. In each round of the simulation, the number of signers participated in the signing were increased by one. In each round, the simulator ran the KGC for signers to extract the partial private-public key pair before process the key validation, signing and verification. A message used in the experiment has been randomly generated in each round with a fixed size of 30 bytes. From the results in Figure 1, 2, 3, 4 and 5, our ACLDVS shows the positive result in every experiment. Even through it cannot surpass the Cui *et al.*'s ALCS scheme in some parts, it was significant compared to the other two schemes and benefited from the designated verifier property over the Cui *et al.*'s ALCS scheme.

VI. CONCLUSION

Privacy issue over the information shared in the cloud storage or in the VANET without an efficient and proper control mechanism has motivated us to provide schemes resolving it. The notion of a aggregatable certificateless designated verifier signature scheme captures the need for the integrity, authenticity, authority, and privacy, which presents as a perfect tool to enable efficient privacy-preserving authentication systems for VANET. Moreover, our ACLDVS signature is aggregatable, which it is helpful in reducing the communication cost in the ad hoc network environment.

REFERENCES

[1] J. A. Akinyele, C. Garman, I. Miers, M. W. Pagano, M. Rushanan, M. Green, and A. D. Rubin, "Charm: A framework for rapidly prototyping cryptosystems," *J. Cryptograph. Eng.*, vol. 3, no. 2, pp. 111–128, Jun. 2013.

[2] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Proc. 9th Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, in Lecture Notes in Computer Science, vol. 2894. Taipei, Taiwan: Springer, Nov./Dec. 2003, pp. 452–473.

[3] M. H. Au, Y. Mu, J. Chen, D. S. Wong, J. K. Liu, and G. Yang, "Malicious KGC attacks in certificateless cryptography," in *Proc. 2nd ACM Symp. Inf., Comput. Commun. Secur. (ASIACCS)*, 2007, pp. 302–311.

[4] M. Bellare and G. Neven, "Multi-signatures in the plain public-key model and a general forking lemma," in *Proc. 13th ACM Conf. Comput. Commun. Secur. (CCS)*, Alexandria, VA, USA, 2006, pp. 390–399.

[5] D. Boneh, "The decision Diffie–Hellman problem," in *Proc. 3rd Int. Symp. Algorithmic Number Theory (ANTS)*, in Lecture Notes in Computer Science, vol. 1423. Portland, OR, USA: Springer-Verlag, Jun. 1998, pp. 48–63.

[6] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, in Lecture Notes in Computer Science, vol. 2656. Warsaw, Poland: Springer, May 2003, pp. 416–432.

[7] S. Canard and V. C. Trinh, "Certificateless public key cryptography in the standard model," *Fundam. Inf.*, vol. 161, no. 3, pp. 219–248, Jul. 2018.

[8] A. De Caro and V. Iovino, "JPBC: Java pairing based cryptography," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Corfu, Greece, Jun. 2011, pp. 850–855.

[9] J. Cui, J. Zhang, H. Zhong, R. Shi, and Y. Xu, "An efficient certificateless aggregate signature without pairings for vehicular ad hoc networks," *Inf. Sci.*, vols. 451–452, pp. 1–15, Jul. 2018.

[10] L. Deng, Y. Yang, and Y. Chen, "Certificateless short aggregate signature scheme for mobile devices," *IEEE Access*, vol. 7, pp. 87162–87168, 2019.

[11] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. IT-22, no. 6, pp. 644–654, Nov. 1976.

[12] S. Goldwasser, S. Micali, and R. L. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM J. Comput.*, vol. 17, no. 2, pp. 281–308, Apr. 1988.

[13] S. Han, M. Xie, B. Yang, R. Lu, H. Bao, J. Lin, H.-B. Hong, M.-X. Gu, and S. Han, "A certificateless verifiable strong designated verifier signature scheme," *IEEE Access*, vol. 7, pp. 126391–126408, 2019.

[14] K. Hashimoto and W. Ogata, "Unrestricted and compact certificateless aggregate signature scheme," *Inf. Sci.*, vol. 487, pp. 97–114, Jun. 2019.

[15] X. Huang, Y. Mu, W. Susilo, D. S. Wong, and W. Wu, "Certificateless signature revisited," in *Proc. 12th Australas. Conf. Inf. Secur. Privacy*, Townsville, QLD, Australia, Jul. 2007, pp. 308–322.

[16] X. Huang, Y. Mu, W. Susilo, and F. Zhang, "Short designated verifier proxy signature from pairings," in *Proc. Workshops Embedded Ubiquitous Comput. (EUC)*, in Lecture Notes in Computer Science, vol. 3823. Nagasaki, Japan: Springer-Verlag, Dec. 2005, pp. 835–844.

[17] X. Huang, W. Susilo, Y. Mu, and F. Zhang, "On the security of certificateless signature schemes from Asiacrypt 2003," in *Proc. 4th Int. Conf. Cryptol. Netw. Secur.*, in Lecture Notes in Computer Science, vol. 3810. Xiamen, China: Springer, Dec. 2005, pp. 13–25.

[18] X. Huang, W. Susilo, Y. Mu, and F. Zhang, "Certificateless designated verifier signature schemes," in *Proc. 20th Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, Vienna, Austria, Apr. 2006, pp. 15–19.

[19] M. Jakobsson, K. Sako, and R. Impagliazzo, "Designated verifier proofs and their applications," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn. (EUROCRYPT)*, in Lecture Notes in Computer Science, vol. 1070. Saragossa, Spain: Springer-Verlag, May 1996, pp. 143–154.

[20] A. Karati, S. H. Islam, and M. Karuppiah, "Provably secure and lightweight certificateless signature scheme for IIoT environments," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3701–3711, Aug. 2018.

[21] F. Laguillaumie and D. Vergnaud, "Multi-designated verifiers signatures: Anonymity without encryption," *Inf. Process. Lett.*, vol. 102, nos. 2–3, pp. 127–132, Apr. 2007.

[22] J. Li, N. Qian, Y. Zhang, and X. Huang, "An efficient certificate-based designated verifier signature scheme," *Comput. Informat.*, vol. 35, no. 5, pp. 1210–1230, 2016.

[23] K. Li, M. H. Au, W. H. Ho, and Y. L. Wang, "An efficient conditional privacy-preserving authentication scheme for vehicular ad hoc networks using online/offline certificateless aggregate signature," in *Proc. 13th Int. Conf. Provable Secur.*, in Lecture Notes in Computer Science, vol. 11821. Cairns, QLD, Australia: Springer, Oct. 2019, pp. 59–76.

[24] H. Lipmaa, G. Wang, and F. Bao, "Designated verifier signature schemes: Attacks, new security notions and a new construction," in *Proc. 32nd Int. Colloq. Automat., Lang. Program. (ICALP)*, in Lecture Notes in Computer Science, vol. 3580. Lisbon, Portugal: Springer-Verlag, Jul. 2005, pp. 459–471.

[25] A. K. Malhi, S. Batra, and H. S. Pannu, "Security of vehicular ad-hoc networks: A comprehensive survey," *Comput. Secur.*, vol. 89, Feb. 2020, Art. no. 101664.

[26] B. Mokhtar and M. Azab, "Survey on security issues in vehicular ad hoc networks," *Alexandria Eng. J.*, vol. 54, no. 4, pp. 1115–1126, Dec. 2015.

[27] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *J. Cryptol.*, vol. 13, no. 3, pp. 361–396, Jun. 2000.

[28] P. Rastegari, W. Susilo, and M. Dakhilalian, "Certificateless designated verifier signature revisited: Achieving a concrete scheme in the standard model," *Int. J. Inf. Secur.*, vol. 18, no. 5, pp. 619–635, Oct. 2019.

[29] S. Sharma and B. Kaushik, "A survey on Internet of vehicles: Applications, security issues & solutions," *Veh. Commun.*, vol. 20, Dec. 2019, Art. no. 100182.

[30] M. S. Sheikh, J. Liang, and W. Wang, "A survey of security services, attacks, and applications for vehicular ad hoc networks (VANETs)," *Sensors*, vol. 19, no. 16, p. 3589, Aug. 2019.

[31] L. Shen, J. Ma, Y. Miao, and H. Liu, "Provably secure certificateless aggregate signature scheme with designated verifier in an improved security model," *IET Inf. Secur.*, vol. 13, no. 3, pp. 167–173, May 2019.
[32] R. Steinfeld, L. Bull, H. Wang, and J. Pieprzyk, "Universal designated-verifier signatures," in *Proc. 9th Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, in Lecture Notes in Computer Science, vol. 2894. Taipei, Taiwan: Springer, Nov./Dec. 2003, pp. 523–542.
[33] W. Susilo, F. Zhang, and Y. Mu, "Identity-based strong designated verifier signature schemes," in *Proc. 9th Australas. Conf. Inf. Secur. Privacy (ACISP)*, in Lecture Notes in Computer Science, vol. 3108. Sydney, NSW, Australia: Springer-Verlag, Jul. 2004, pp. 313–324.
[34] P. Thorncharoensri, W. Susilo, and Y. Mu, "How to balance privacy with authenticity," in *Proc. 11th Int. Conf. Inf. Secur. Cryptol.*, in Lecture Notes in Computer Science, vol. 5461. Seoul, South Korea: Springer, Dec. 2008, pp. 184–201.
[35] P. Thorncharoensri, W. Susilo, and Y. Mu, "Policy-controlled signatures," in *Proc. 11th Int. Conf. Inf. Commun. Secur.*, in Lecture Notes in Computer Science, vol. 5927. Beijing, China: Springer, Dec. 2009, pp. 91–106.
[36] P. Thorncharoensri, W. Susilo, and Y. Mu, "Universal designated verifier signatures with threshold-signers," in *Proc. 4th Int. Workshop Secur. (IWSEC)*, in Lecture Notes in Computer Science, vol. 5824. Toyama, Japan: Springer, Oct. 2009, pp. 89–109.
[37] P. Thorncharoensri, W. Susilo, and Y. Mu, "Policy controlled system with anonymity," *Theor. Comput. Sci.*, vol. 745, pp. 87–113, Oct. 2018.
[38] W. Yang, S. Wang, X. Huang, and Y. Mu, "On the security of an efficient and robust certificateless signature scheme for IIoT environments," *IEEE Access*, vol. 7, pp. 91074–91079, 2019.
[39] Y. Zhang, R. H. Deng, D. Zheng, J. Li, P. Wu, and J. Cao, "Efficient and robust certificateless signature for data crowdsensing in cloud-assisted industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 15, no. 9, pp. 5099–5108, Sep. 2019.

PAIRAT THORNCHAROENSRI received the bachelor's degree in electrical engineering from the King Mongkut's University of Technology North Bangkok, Thailand, and the M.Sc. degree in computer science and the Ph.D. degree from the University of Wollongong (UOW), Australia. He is a Lecturer with the School of Computer Science and Information Technology and a member of the Institute of Cybersecurity and Cryptology, UOW. His main research interests are currently focused on privacy preservative techniques in the Internet of Things, blockchain, cloud computing, and big data.

WILLY SUSILO (Senior Member, IEEE) received the Ph.D. degree from the University of Wollongong (UOW), Australia, in 2001. He has been the Head of the School of Computing and Information Technology, UOW, since 2015. He was the former Head of the School of Computer Science and Software Engineering, from 2009 to 2010, and the Deputy Director of the ICT Research Institute, UOW, from 2006 to 2008. He is an innovative educator and researcher. He is a Senior Professor with the School of Computing and Information Technology, Faculty of Engineering and Information Sciences, UOW. He is the Director of Institute of Cybersecurity and Cryptology, UOW. He has published more than 300 articles in journals and conference proceedings in cryptography and network security. He has served as a program committee member of several international conferences. He was awarded the prestigious Australian Research Council Future Fellowship, in 2009. In 2016, he was awarded the Researcher of the Year from the UOW, for his research excellence and contributions. His work on the creation of short signature schemes has been well cited and it is a part of the IETF draft. He is the Editor-in-Chief of the *Information* journal.

JOONSANG BAEK received the Ph.D. degree from Monash University, Australia, in 2004. His Ph.D. thesis was on the security analysis of signcryption. He has received great attention from the research community. He was a Research Scientist with the Institute for Infocomm Research, Singapore, and an Assistant Professor with the Khalifa University of Science and Technology, United Arab Emirates. He is a Senior Lecturer with the School of Computer Science and Information Technology and a member of the Institute of Cybersecurity and Cryptology, University of Wollongong (UOW), Australia. He has published his work in numerous reputed journals and conference proceedings. His current research interests are in the fields of applied cryptography and cybersecurity. He has also served as a program committee member and the chair of a number of renowned conferences on information security and cryptography.