International Journal of Information Security (2019) 18:619–635
https://doi.org/10.1007/s10207-019-00430-5

REGULAR CONTRIBUTION

Certificateless designated verifier signature revisited: achieving a concrete scheme in the standard model

Parvin Rastegari[1] · Willy Susilo[2] · Mohammad Dakhilalian[1]

Published online: 27 March 2019

Abstract
In a designated verifier signature (DVS) scheme, the signer (Alice) creates a signature which is only verifiable by a designated verifier (Bob). Furthermore, Bob cannot convince any third party that the signature was produced by Alice. A DVS scheme is applicable in scenarios where Alice must be authenticated to Bob without disturbing her privacy. The de-facto construction of DVS scheme is achieved in a traditional public key infrastructure (PKI) setting, which unfortunately requires a high-cost certificate management. A variant of identity-based (ID-based) setting DVS eliminates the need of certificates, but it introduces a new inherent key escrow problem, which makes it impractical. Certificateless public key cryptography (CL-PKC) is empowered to overcome the problems of PKI and ID-based settings, where it does not suffer from any of the aforementioned problems. However, only a few number of certificateless DVS (CL-DVS) schemes have been proposed in the literature to date. Moreover, all existing CL-DVS schemes are only proven secure in the random oracle model, while some of them are already known to be insecure. We provide three contributions in this paper. First, we revisit the security proofs of existing CL-DVS schemes in the literature and show that unfortunately there are some drawbacks in the proofs of all of those schemes. Second, we concentrate on the recently proposed CL-DVS scheme (IEEE Access 2018) and show a drawback in its security proof which makes it unreliable. Furthermore, we show that this scheme is delegatable in contrast to the author's claim. Finally, we propose a CL-DVS scheme and prove its security requirements *in the standard model*. Our scheme is not only *the first scheme* with a complete and correct security proofs, but also *the only scheme in the standard model*.

Keywords Designated verifier signature · Certificateless public key cryptography · Certificateless designated verifier signature · Standard model · Random oracle model

1 Introduction

A digital signature is a well-known primitive that provides integrity and authenticity of messages in cryptographic protocols [1]. In an ordinary digital signature scheme, the signer (Alice) creates a signature which is publicly verifiable by everyone. The privacy of Alice is not preserved in a traditional digital signature, since a verifier (Bob) can convince any third party about Alice's signature by presenting her signature to the third party (without the need of Alice's consent). This public verifiability of digital signatures is a useful and necessary property in some applications, but it is not a desirable property in some specific applications such as e-votings, e-auctions and fair exchanges, in which integrity and authenticity must be satisfied without disturbing the privacy of the signer. Many works have been devoted to overcome the conflicts between the authenticity and privacy of the signer in digital signatures. In 1989, the concept of undeniable signature was proposed in which some help of the signer is necessary in the verification phase [2]. In 1996, Jakobsson et al. [3] and Chaum [4] independently proposed the concept of designated verifier signature/proof (DVS/DVP), in order to avoid the interaction between the signer and the verifier. In a DVS scheme, a signer (Alice) generates a signature

✉ Willy Susilo
wsusilo@uow.edu.au

Parvin Rastegari
parvin.rastegari@ec.iut.ac.ir

Mohammad Dakhilalian
mdalian@cc.iut.ac.ir

[1] Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan 84156-83111, Iran

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

which can only be verified by a designated verifier (Bob). Moreover, Bob cannot transfer his conviction about Alice's signature to any third party, since he can produce a signature which is indistinguishable from the one generated by Alice. Thus, the authenticity of Alice is proved to Bob and her privacy is preserved at the same time, without any interaction between Alice and Bob. Note that the main advantage of a DVS scheme in comparison with other privacy-preserving authentication schemes such as [5,6] is privacy-preserved authentication of Alice to Bob without any reciprocal communications between Alice and Bob.

In a traditional public key cryptography (PKC), a user selects a public/private key pair (PK, SK) for him/herself. In a conventional public key infrastructure (PKI), a digital certificate is issued by a certificate authority (CA) to bind between the public key and the identity of a user. The management of the certificates requires a large amount of computation, storage and communication costs in a traditional PKI. To solve this problem, the notion of identity-based cryptography (ID-PKC) was put forth by Shamir in 1984 [7]. In an ID-PKC, a trusted third party called the private key generator (PKG) produces the private key of a user from his/her unique identifier information. An inherent problem of ID-PKC is the key escrow problem, i.e., the PKG knows all users' private keys. To overcome these problems simultaneously, Al Riyami and Paterson introduced the concept of certificateless public key cryptography (CL-PKC) in 2003 [8]. In a CL-PKC, a public/secret key pair (PK, x) is generated by the user him/herself without requiring PK to be certified. Also, a partial private key d is created by a semi-trusted third party called key generation center (KGC), from the unique identifier information of the user. The knowledge of both x and d is required for a user to acquire his/her full private key SK. One can consider CL-PKC as an intermediate solution between a traditional PKI and ID-PKC.

The concept of certificateless designated verifier signature (CL-DVS) was put forward by Huang et al. [9]. Subsequently, only several CL-DVS schemes have been proposed in the literature [10–17]. Although these schemes are claimed to be secure in the random oracle model (ROM), unfortunately there exist some drawbacks in the security proofs of all of them. Additionally, the proposed scheme in [16] is insecure against the key replacement attack proposed in [18]. Furthermore, since the introduction of the notion of the malicious KGC attack in CL-PKC [19], the need for a secure scheme under this type of attack becomes essential. Unfortunately, as shown in [19,20], the proposed CL-DVS schemes in [9,14] are insecure against malicious KGC attack. There are also key-compromise and malicious KGC attacks against the scheme in [14] proposed in [21]. We should note that the malicious KGC attack was not claimed in the original paper of [9], but the work in [19] shows that there is some lack of completeness of this paper in accordance with the current state-of-the-art knowledge.

Our Contributions

We provide three contributions in this paper. First, we revisit the existing CL-DVS schemes and their security proofs and show that unfortunately there are issues in the security proofs of all of those schemes. Second, we concentrate on the recently proposed CL-DVS scheme [17] and show some drawbacks in its security proofs which make them unreliable. Moreover, we show that this scheme is delegatable in contrast to the author's claim. Finally, we present a CL-DVS scheme and prove its security requirements *without random oracles*. Note that as Rogaway discussed in [22], the schemes which their security requirements are proved in the ROM might not be secure when the random oracles are replaced with the real-world primitives (such as hash functions). Therefore, we provide the security proof of our proposal in the standard model (without random oracles). Not only our proposal is *the first CL-DVS scheme in the standard model*, but also it does not suffer from the drawbacks in the security proofs of the previous proposals.

Paper Organization

In Sect. 2, we provide some preliminaries that will be used throughout this work. In Sect. 3, the formal model of a CL-DVS scheme is described. In Sect. 4, we provide a discussion on the security proofs of CL-DVS schemes. In Sect. 5, we provide a discussion on a recently proposed CL-DVS scheme (IEEE Access 2018) and show some drawbacks in this scheme. Subsequently, in Sect. 6, we propose our new CL-DVS scheme. In Sect. 7, we prove the security requirements of our proposal in the standard model. In Sect. 8, a comparison between our proposal and other existing schemes is provided. Finally, Sect. 9 concludes this work.

2 Preliminaries

2.1 Bilinear pairings

Consider two multiplicative cyclic groups G_1 and G_2 of prime order p and let g be a generator of G_1. The mapping $e : G_1 \times G_1 \longrightarrow G_2$ is an admissible bilinear pairing if and only if the following properties are satisfied:

1. Bilinearity: $e(g^a, g^b) = e(g, g)^{ab}$, for all $a, b \in \mathbb{Z}_p^*$.
2. Non-degeneracy: i. e., $e(g, g) \neq 1_{G_2}$.
3. Computability: There exists an efficient algorithm for computing $e(g, g)$.

The readers can refer to [23] for more details about bilinear pairings.

2.2 Related complexity assumption

Consider a set of integers $\mathcal{S} \subset \mathbb{Z}$, and define $\mathcal{S} +_p \mathcal{S} \triangleq \{i + j \bmod \lambda(p) : i, j \in \mathcal{S}\}$, where $\lambda(p)$ is the order of elements modulo p. Also, consider another target integer $m \notin S +_p \mathcal{S}$. The $(\mathcal{S}, m)$-Computational-Bilinear Diffie–Hellman Exponent-Set ($(\mathcal{S}, m)$-CBDHE-Set) problem [24] is that on inputs $\{g^{a^i} \in G_1 : i \in \mathcal{S}\}$, for unknown $a \in \mathbb{Z}_p^*$, calculate $e(g, g)^{a^m}$. It is said that (ε, t)-$(\mathcal{S}, m)$-CBDHE-Set assumption holds in (G_1, G_2), if no t-time algorithm can solve the $(\mathcal{S}, m)$-CBDHE-Set problem in (G_1, G_2), with probability at least ε.

Remark 1 By assigning $\mathcal{S} = \mathcal{S}_K = \{0, 1, 2, \ldots, K\}$, and $m = 2K + 1$, the $(\mathcal{S}, m)$-CBDHE-Set problem is that on inputs $g, g^a, g^{a^2}, \ldots, g^{a^K} \in G_1$, for unknown $a \in \mathbb{Z}_p^*$, calculate the value of $e(g, g)^{a^{2K+1}}$. The unforgeability of our proposed scheme is based on $(\mathcal{S}_2, 5)$-CBDHE-Set and $(\mathcal{S}_4, 9)$-CBDHE-Set assumptions against $\mathcal{A}_I$ and $\mathcal{A}_{II}$, respectively.

3 Formal model of CL-DVS

3.1 Algorithms and syntax

A CL-DVS scheme comprises three entities: the key generation center (KGC), the real signer (S) and the designated verifier (V). It is defined by the following eight algorithms [9]:

- **Setup** KGC runs this probabilistic polynomial time (PPT) algorithm which takes a security parameter λ as input and outputs system parameters $params$ and a master secret key msk. Then, KGC publishes $params$ and keeps msk secret.
- **Partial-Private-Key-Extract (PPKE)** KGC runs this PPT algorithm which takes $params$, msk and an identity $ID_U \in \{0, 1\}^*$ as input and outputs a partial private key d_U. Then KGC sends d_U to the corresponding user via a secure channel.
- **Set-Secret-Value (SSV)** The user with identity ID_U runs this PPT algorithm which takes $params$ and ID_U as input and outputs a random choice x_U. The user keeps x_U as his secret value.
- **Set-Private-Key (SPrK)** The user with identity ID_U runs this (P)PT algorithm which takes $params$, ID_U, d_U and x_U as input and outputs the user's full private key SK_U.
- **Set-Public-Key (SPuK)** The user with identity ID_U runs this PT algorithm which takes $params$, ID_U, x_U (and maybe d_U) as input and outputs the user's public key PK_U. The user publishes PK_U without requiring to be certified.
- **DVS Signing (DSign)** The signer with identity ID_S runs this (P)PT algorithm which takes $params$, a message m, the signer's identity ID_S, the signer's private key SK_S, the verifier's identity ID_V and the verifier's public key PK_V as input and outputs a signature σ on message m.
- **DVS Verification (DVer)** The verifier with identity ID_V runs this deterministic PT algorithm which takes $params$, a message/DVS pair (m, σ), ID_S, PK_S, ID_V and SK_V as input and outputs 1 if σ is valid and 0, otherwise.
- **Transcript Simulation (TS)** The verifier with identity ID_V is able to run this algorithm to produce a signature σ' which is indistinguishable from σ created by the original signer.

Remark 2 The SPrK algorithm is defined based on one of the two following methods:

- **SPrK1** SK_U is directly set as x_U and d_U (or maybe a part of d_U), i.e., $SK_U = (d_U, x_U)$, such as the schemes in [10,11,14–17].
- **SPrK2** SK_U is set as a function of d_U and x_U, i.e., $SK_U = func(d_U, x_U)$, such as the scheme in [9].

There are some subtle differences in oracle accesses in the security proof of CL-DVS schemes based on whether they use SPrK1 or SPrK2 algorithms, which will be discussed in Sect. 4.

3.2 Security requirements

Correctness, unforgeability and non-transferability (source hiding) are three basic security requirements of a CL-DVS scheme. They are described as follows.

Correctness If the signer properly creates a CL-DVS by the DSign algorithm, then this signature must pass the DVer algorithm successfully.
Unforgeability It is computationally infeasible to create a valid CL-DVS for everyone except the signer or the designated verifier. See Sect. 4 for more descriptions about the security model for proving the unforgeability of a CL-DVS scheme.
Non-Transferability Given a message m and a CL-DVS σ on m, it is infeasible to determine whether the original signer or the designated verifier creates σ, even if one knows all private keys.

Remark 3 In 2005, a new security notion called as non-delegatability was proposed for designated verifier signature schemes [25]. According to this property, neither the signer

nor the designated verifier is able to delegate the signing rights to a third party without revealing his/her private key. However, the authors in [26] pointed out that although the non-delegatability has been a focus of many recent researches, it may be undesirable in some applications. Therefore, the non-delegatable DVS schemes should be considered as a special category which are useful in specific applications where the responsibility of the signer is important and cannot be delegated to another entity. Among all proposed CL-DVS schemes [9–17], only the schemes in [15–17] are claimed to be non-delegatable. However, the scheme in [16] is forgeable according to the attack proposed in [18]. Moreover, we show in Sect. 5.1 that the scheme in [17] is delegatable in contrast to its author's claim. We should emphasize that our proposal (in Sect. 6) is not placed in the category of non-delegatable CL-DVS schemes, too.

4 A discussion on security proofs

In a certificateless public key cryptography, two types of adversaries are considered [9–17]:

1. The type I adversary $\mathcal{A}_I$ who cannot obtain the master secret key but he can replace the public key of an arbitrary entity because of the uncertified nature of the public keys generated by the users. In fact, $\mathcal{A}_I$ is a key replacement attacker.
2. The type II adversary $\mathcal{A}_{II}$ who possesses the master secret key but cannot replace any public keys (malicious KGC attacker).

In a CL-DVS scheme, the adversaries may access to the following oracles:

Hash Oracle ($\mathcal{O}_H$): Given any input, returns its hash function.
PPKE Oracle ($\mathcal{O}_d$): Given an identity ID_U, returns d_U.
SPuK Oracle ($\mathcal{O}_{PK}$): Given an identity ID_U, returns PK_U.
SSV Oracle ($\mathcal{O}_x$): Given an identity ID_U, returns x_U.
SPrK Oracle ($\mathcal{O}_{SK}$): Given an identity ID_U, returns SK_U.
Replace-Public-Key Oracle ($\mathcal{O}_{RPK}$): Given an identity ID_U and a new valid public key PK'_U, replaces PK_U with PK'_U.
DSign Oracle ($\mathcal{O}_S$): Given a message m, ID_S and ID_V, returns a valid signature σ on m from S to V.
DVer ($\mathcal{O}_V$): Given a signature σ, ID_S and ID_V, returns 1 if σ is valid and 0 otherwise.

The third and the forth columns of Table 1 show the oracle accesses of $\mathcal{A}_I$ and $\mathcal{A}_{II}$ in the proposed CL-DVS schemes, while the second column shows whether the scheme is based on SPrK1 or SPrK2 algorithms (which is described in Sect. 3.1). Note that $\mathcal{O}_I$ and $\mathcal{O}_{II}$ denote the set of oracles which can be accessed by $\mathcal{A}_I$ and $\mathcal{A}_{II}$, respectively.

Unforgeability of a CL-DVS scheme is defined by the two following games for type I and type II adversaries, respectively.

Game 1 *A challenger $\mathcal{C}$ plays this game with $\mathcal{A}_I$ as follows:*

- ***Setup*** *$\mathcal{C}$ takes a security parameter λ as input and produces params and msk. $\mathcal{C}$ sends params to $\mathcal{A}_I$ and keeps msk secret.*
- ***Queries*** *$\mathcal{A}_I$ issues polynomially bounded number of queries to the oracles in set $\mathcal{O}_I$ and $\mathcal{C}$ must answer these queries by simulating the oracles.*
- ***Forgery*** *Finally, $\mathcal{A}_I$ creates a forged message/signature (m^*, σ^*) from the signer with identity ID_{S^*} to the verifier with identity ID_{V^*}.*

Game 2 *A challenger $\mathcal{C}$ plays this game with $\mathcal{A}_{II}$ as follows:*

- ***Setup*** *$\mathcal{C}$ takes a security parameter λ as input and produces params and msk. $\mathcal{C}$ sends params and msk to $\mathcal{A}_{II}$.*
- ***Queries*** *$\mathcal{A}_{II}$ issues polynomially bounded number of queries to the oracles in set $\mathcal{O}_{II}$ and $\mathcal{C}$ must answer these queries by simulating the oracles.*
- ***Forgery*** *Finally, $\mathcal{A}_{II}$ creates a forged message/signature (m^*, σ^*) from the signer with identity ID_{S^*} to the verifier with identity ID_{V^*}.*

It is said that $\mathcal{A}_I$ wins Game 1 (or $\mathcal{A}_{II}$ wins Game 2) if σ^* is a valid signature on m^* with respect to ID_{S^*} and ID_{V^*} and meanwhile σ^* is not obtained from the $\mathcal{O}_{DSign}$ oracle. Moreover there are some trivial conditions that must be satisfied according to the oracle accesses and the construction of the scheme which will be discussed in Sect. 4.2.

4.1 A discussion on oracle accesses

Here, we provide a discussion on oracle accesses in the proposed CL-DVS schemes which are shown in Table 1. Note that the adversaries may obtain some information from the environment which must be simulated by the oracle accesses in the security model. As the authors in [11] do not model the oracle accesses, their security proof is not rigorous at all. So we do not consider this scheme in the following discussions.

When the security is proved in the ROM, the adversaries are considered to have access to $\mathcal{O}_H$. In fact, a hash function is simulated as a random oracle and as Rogaway discussed in [22], these schemes are not secure when the random oracles are replaced with the real hash functions. As the proposed

Table 1 Oracle accesses in the existing CL-DVS schemes

Scheme	SPrK	$\mathcal{O}_I$	$\mathcal{O}_{II}$
[9]	SPrK2	$\{\mathcal{O}_H, \mathcal{O}_S, \mathcal{O}_V\}$	$\{\mathcal{O}_H, \mathcal{O}_S, \mathcal{O}_V\}$
[10]	SPrK1	$\{\mathcal{O}_H, \mathcal{O}_S, \mathcal{O}_V\}$	$\{\mathcal{O}_H, \mathcal{O}_S, \mathcal{O}_V\}$
[11]	SPrK1	$\times$	$\times$
[12]	SPrK1	$\{\mathcal{O}_H, \mathcal{O}_d, \mathcal{O}_{PK}, \mathcal{O}_x, \mathcal{O}_{RPK}, \mathcal{O}_S, \mathcal{O}_V\}$	$\{\mathcal{O}_H, \mathcal{O}_{PK}, \mathcal{O}_x, \mathcal{O}_S, \mathcal{O}_V\}$
[13]	SPrK1	$\{\mathcal{O}_H, \mathcal{O}_d, \mathcal{O}_{PK}, \mathcal{O}_x, \mathcal{O}_{RPK}, \mathcal{O}_S, \mathcal{O}_V\}$	$\{\mathcal{O}_H, \mathcal{O}_{PK}, \mathcal{O}_x, \mathcal{O}_S, \mathcal{O}_V\}$
[14]	SPrK1	$\{\mathcal{O}_H, \mathcal{O}_d, \mathcal{O}_{PK}, \mathcal{O}_x, \mathcal{O}_{RPK}, \mathcal{O}_S, \mathcal{O}_V\}$	$\{\mathcal{O}_H, \mathcal{O}_{PK}, \mathcal{O}_x, \mathcal{O}_S, \mathcal{O}_V\}$
[15]	SPrK1	$\{\mathcal{O}_H, \mathcal{O}_d, \mathcal{O}_{PK}, \mathcal{O}_{SK}, \mathcal{O}_{RPK}, \mathcal{O}_S\}$	$\{\mathcal{O}_H, \mathcal{O}_d, \mathcal{O}_{PK}, \mathcal{O}_{SK}, \mathcal{O}_S\}$
[16]	SPrK1	$\{\mathcal{O}_H, \mathcal{O}_d, \mathcal{O}_{PK}, \mathcal{O}_x, \mathcal{O}_{RPK}, \mathcal{O}_S, \mathcal{O}_V\}$	$\{\mathcal{O}_H, \mathcal{O}_{PK}, \mathcal{O}_x, \mathcal{O}_S, \mathcal{O}_V\}$
[17]	SPrK1	$\{\mathcal{O}_H, \mathcal{O}_d, \mathcal{O}_{PK}, \mathcal{O}_x, \mathcal{O}_{RPK}, \mathcal{O}_S, \mathcal{O}_V\}$	$\{\mathcal{O}_H, \mathcal{O}_d, \mathcal{O}_{PK}, \mathcal{O}_x, \mathcal{O}_S, \mathcal{O}_V\}$

CL-DVS schemes in [9,10,12–17] are provable in the ROM, the adversaries have access to $\mathcal{O}_H$ in the security proof of all of them.

In the security model of the schemes in [9,10], the adversaries have only access to $\{\mathcal{O}_H, \mathcal{O}_S, \mathcal{O}_V\}$. In fact, in these schemes, only the existential unforgeability against chosen message attack (EUF-CMA) is considered in which the adversaries can only obtain a signature on every message m and the verification result on every signature from ID_{S^*} to ID_{V^*} and finally forge a signature on a new challenge message m^* from ID_{S^*} to ID_{V^*}. However, in a certificateless setting, the existential unforgeability against chosen message and identity attack (EUF-CM&IDA) must be considered in which the adversaries are considered more powerful as they can obtain the signature on every message m from every ID_S to every ID_V, the verification result of every signature from every ID_S to every ID_V, and even the keys of every ID_U of their choice and finally forge a signature on a challenge message m^* from ID_S^* to ID_V^*. Hence, we concentrate on the security model of the schemes in [12–17] in which the EUF-CM&IDA is considered.

For considering EUF-CM&IDA, $\mathcal{A}_I$ has access to the oracles in set $\mathcal{O}_I = \{\mathcal{O}_H, \mathcal{O}_d, \mathcal{O}_{PK}, \mathcal{O}_x$(and/or $\mathcal{O}_{SK}$), $\mathcal{O}_{RPK}, \mathcal{O}_S, \mathcal{O}_V\}$. However, $\mathcal{A}_{II}$ (a malicious KGC) possesses the master secret key and is able to obtain d_U of every user, so he/she does not require to have access to $\mathcal{O}_d$ (while it is not any problem to consider $\mathcal{O}_d$ in set $\mathcal{O}_{II}$ such as [15,17]). Moreover, it is assumed that $\mathcal{A}_{II}$ does not replace public keys and so he/she does not have access to $\mathcal{O}_{RPK}$, too. As a result, $\mathcal{A}_{II}$ has access to the oracles in set $\mathcal{O}_{II} = \{\mathcal{O}_H, \mathcal{O}_{PK}, \mathcal{O}_x$(and/or $\mathcal{O}_{SK}$), $\mathcal{O}_S, \mathcal{O}_V\}$.

As shown in Table 1, all of the schemes in [12–17] are based on SPrK1 algorithms, i.e., $SK_U = (d_U, x_U)$, and the adversary can obtain SK_U by obtaining x_U from $\mathcal{O}_x$ and d_U from $\mathcal{O}_d$, so it is not a requirement for simulating $\mathcal{O}_{SK}$ in these schemes. However, in the scheme in [15], only $\mathcal{O}_{SK}$ is considered which is in fact a combination of $\mathcal{O}_x$ and $\mathcal{O}_d$. In CL-DVS schemes based on the SPrK2 algorithm, $SK_U = func(x_U, d_U)$, i.e., x_U is not directly used in the signing process and so it is sufficient to consider $\mathcal{O}_{SK}$ without simulating $\mathcal{O}_x$ [27].

As shown in Table 1, in the security proof of the scheme in [15], $\mathcal{O}_V$ is not simulated, so its security proof is unfortunately not rigorous as well. As a result, only in the proofs of the schemes in [12–14,16,17] the oracle accesses seems to be reasonable.

4.2 The conditions for the adversary success

As mentioned, it is said that $\mathcal{A}_I$ wins Game 1 (or $\mathcal{A}_{II}$ wins Game 2) if σ^* is a valid signature on m^* with respect to ID_{S^*} and ID_{V^*} and meanwhile σ^* is not obtained from the $\mathcal{O}_{DSign}$ oracle. Furthermore, there are some trivial conditions that must be satisfied according to the construction of the scheme and the oracle accesses. In this section, we provide a discussion on these conditions. We only consider the schemes in which EUF-CM&IDA is considered (i.e., the schemes in [12–17]).

It is obvious that SK_{S^*} must not be extracted at any point, since it is trivial that by obtaining SK_{S^*}, the adversary can forge a signature by running the DSign algorithm. Moreover, as in all of the schemes in [12–17], the verifier can use his full private key SK_V to create a valid signature by the TS algorithm, SK_{V^*} must not be extracted at any point, too.

There are four conditions for the success of the adversary ($\mathcal{A}_I$ in Game 1 or $\mathcal{A}_{II}$ in Game 2) in the proposed schemes as follows:

- C1: ID_{S^*} and ID_{V^*} have not been submitted to $\mathcal{O}_d$.
- C2: ID_{S^*} and ID_{V^*} have not been submitted to $\mathcal{O}_x$.
- C3: ID_{S^*} has not been submitted to $\mathcal{O}_{RPK}$.
- C4: ID_{V^*} has not been submitted to $\mathcal{O}_{RPK}$.

There are some contradictions between the conditions which are considered in the definition and in the process of the proof for some of the proposed schemes. In Table 2, the conditions for winning $\mathcal{A}_I$ in Game 1 and $\mathcal{A}_{II}$ in Game 2 are summarized both in the definition and the process of the proof

Table 2 The conditions for the success of $\mathcal{A}_I$ and $\mathcal{A}_{II}$

Scheme	The conditions in Game 1		The conditions in Game 2	
	Definition	Proof	Definition	Proof
[12]	?	C1	?	C2
[13]	C1	C1	C2	C2
[14]	C1,C2	C1,C2	C1,C2,C3,C4	C2
[15]	C1	C1,C4	C1	C2
[16]	C1,C2,C3,C4	C1	C1,C2	C2
[17]	C1,C2	C1,C2,C3,C4	C2,C3,C4	C2

? There are not any descriptions for the conditions of the success of adversaries in definitions of [12]

for the schemes in [12–17]. There are some considerations on Table 2 which will be described in Sects. 4.2.1 and 4.2.2.

4.2.1 The conditions for the success of $\mathcal{A}_I$ In Game 1

The conditions C1 and C2 are considered for preventing $\mathcal{A}_I$ from obtaining SK_{S^*} and SK_{V^*}. Note that if $\mathcal{A}_I$ cannot obtain d_{S^*} (d_{V^*}) he/she cannot obtain SK_{S^*} (SK_{V^*}) even if he/she knows x_{S^*} (x_{V^*}). Hence, the condition C1 seems to be sufficient and there is not any reason for considering the condition C2. In other words, C1 must be satisfied in Game 1, but there is not any necessity for satisfying C2, although there is not any problem to consider C1 and C2 together.

About the condition C3, note that $\mathcal{A}_I$ is a key replacement attacker who tries to forge a signature from a signer with identity ID_{S^*} to the verifier with identity ID_{V^*} via Game 1. Hence, it is not reasonable to prevent $\mathcal{A}_I$ from replacing the public key of ID_{S^*}, i.e., C3 is not a reasonable condition at all. Although the authors in [16] consider C3 in their definition of the success of $\mathcal{A}_I$ in Game 1, they do not consider this condition in their proof, so although their definition is not correct, their proof seems to be true in this sense. The only scheme for which this unreasonable condition (i.e., C3) is considered in the security proof is the scheme in [17], so its security proof against $\mathcal{A}_I$ is not rigorous at all (see Sect. 5.2 for more descriptions).

Although C3 is not reasonable, there is not any problem to consider C4 in the conditions of the $\mathcal{A}_I$'s success, since $\mathcal{A}_I$ does not gain any benefits from replacing PK_{V^*}. More precisely, note that $\mathcal{A}_I$ tries to forge a signature from ID_{S^*} to ID_{V^*} via Game 1, i.e., $\mathcal{A}_I$ is trying to cheat ID_{V^*} by a forged signature from ID_{S^*}. Even if $\mathcal{A}_I$ has the permission to replace PK_{V^*} with another public key PK'_{V^*} in Game 1 and forges a signature σ^*, he/she does not obtain any advantage from this replacement, as ID_{V^*} uses his/her real public key PK_{V^*} to verify the forged signature σ^*. As a result, σ^* will not pass the DVer algorithm as it is a forged signature corresponding to PK'_{V^*} not PK_{V^*}. So, as $\mathcal{A}_I$ does not gain any benefits from replacing PK_{V^*}, it is not unreasonable to consider C4 in the conditions of his/her success.

To sum up, considering C1, C2 and C4 is reasonable, while considering C3 is not reasonable for the success of $\mathcal{A}_I$ in Game 1. Hence, as shown in 2, the conditions for the success of $\mathcal{A}_I$ in Game 1 are reasonable in the proof of all of the schemes except the scheme in [17].

4.2.2 The conditions for the success of $\mathcal{A}_{II}$ in Game 2

Note that $\mathcal{A}_{II}$ is a malicious KGC attacker who possesses the master secret key, so he/she can compute the partial private key of every user and does not require to have access to $\mathcal{O}_d$. Moreover, $\mathcal{A}_{II}$ does not have access to $\mathcal{O}_{RPK}$ for any user. As a result the conditions C1, C3 and C4 are meaningless for $\mathcal{A}_{II}$'s success in Game 2. Although these conditions are considered in the definition of $\mathcal{A}_{II}$'s success in some papers (see Table 2), they are not considered in the process of the proof. Therefore, the only reasonable condition for the success of $\mathcal{A}_{II}$ in Game 2 is C2, as if $\mathcal{A}_{II}$ can obtain x_{S^*} (x_{V^*}) he/she can easily obtain SK_{S^*} (SK_{V^*}) as he/she knows all partial private keys. Hence, as shown in 2, the conditions for the success of $\mathcal{A}_{II}$ in Game 2 are reasonable in the proof of all of the schemes.

4.3 Probability analysis

It is well known that in a security proof, a simulator $\mathcal{B}$ is constructed which uses the adversary ($\mathcal{A}_I$ or $\mathcal{A}_{II}$) as a subroutine and tries to solve a hard problem. $\mathcal{B}$ must simulate $\mathcal{C}$ and all oracle accesses of $\mathcal{A}_I$ ($\mathcal{A}_{II}$) in Game 1 (Game 2). If the success probability of the adversary in his/her attack (ε) is non-negligible, then the success probability of $\mathcal{B}$ for solving the hard problem (ε') must be non-negligible too, as would be a contradiction to the intractability of the hard problem. From this contradiction, we conclude that ε is negligible. It is obvious that if ε' is negligible, there is not any contradiction which leads us to conclude that ε is negligible, and so a security proof is valueless without probability analysis. Unfortunately, probability analysis is not provided in none of the schemes in [12–17], except [15,17].

5 Discussions on Lin's scheme

Recently (in IEEE Access 2018), Lin proposed a CL-DVS scheme in the ROM with the claim of satisfying the following properties [17]:

1. Essential security properties, i.e., correctness, unforgeability (against $\mathcal{A}_I$ and $\mathcal{A}_{II}$) and non-transferability which are described in Sect. 4.
2. SSA-KCA (signer ambiguity against key-compromise attack) which is a notion introduced by Lin himself [17].
3. Non-delegatability which is described in Remark 3.

In this section, we first provide an overview on the algorithms of Lin's Scheme. Then, we show that in contrast to his claim, his scheme is delegatable. Moreover, we have a comment on the proof of the unforgeability of his scheme.

5.1 An overview of Lin's scheme

The algorithms of Lin's CL-DVS scheme are as follows:

- **Setup** On input a security parameter λ, the KGC chooses a large prime p and an elliptic curve E/F_p over the field F_p. Let G_1 be a cyclic additive group on E/F_p and P be a base point of order q (another large prime) over G_1. The KGC also picks three collision-resistant hash functions $H_1 : G_1 \times \{0,1\}^* \longrightarrow \mathbb{Z}_q^*$, $H_2 : G_1^4 \longrightarrow \mathbb{Z}_q^*$ and $H_3 : \{0,1\}^* \times G_1 \longrightarrow G_1$. Moreover, the KGC selects a random $s \in_R \mathbb{Z}_q^*$ as his/her master secret key and computes $P_s = sP$ as his/her public key. The public parameters are $\Omega = \{F_p, E/F_p, G_1, q, P, H_1, H_2, H_3, P_s\}$.
- **PPKE** On input an identity ID_i (corresponding to the user U_i), the KGC first selects a random $u_i \in_R \mathbb{Z}_q^*$ and then computes $D_i = u_i P$, $q_i = H_1(D_i, ID_i)$ and $s_i = u_i + q_i s \bmod q$. Then, the partial private key $S_i = (s_i, q_i, D_i)$ is returned to the user U_i. Each user can verify it by checking whether the equality $s_i P = D_i + q_i P_s$ holds or not.
- **SSV** The user U_i, selects a random $x_i \in_R \mathbb{Z}_q^*$ as his secret value and compute $Y_i = x_i P$.
- **SPrK** U_i sets $SK_i = (x_i, s_i)$ as his/her full private key.
- **SPuK** U_i sets $PK_i = (Y_i, q_i, D_i)$ as his/her public key.
- **DSign** Suppose that a signer U_a wants to create a signature on a message m for a designated verifier U_v. The signer selects random values $w, r \in_R \mathbb{Z}_q^*$ and computes:

$$
\begin{aligned}
&l = w + x_a, \quad K = ls_a(D_v + q_v P_s), \quad U = H_3(m, K),\\
&R = rP - U = (r_x, r_y), \quad V = (r + w)Y_v,\\
&Z = r_y x_a Y_v, \quad h = H_2(U, R, Z, V).
\end{aligned}
$$

  Then returns $\sigma = (l, R, h)$ to U_v.

- **DVer** Upon receiving $\sigma = (l, R, h)$, U_v computes:

$$
\begin{aligned}
&W = lP - Y_a, \quad K' = ls_v(D_a + q_a P_s),\\
&U' = H_3(m, K'),\\
&V' = x_v(R + U' + W), \quad Z' = r_y x_v Y_a,\\
&h' = H_2(U', R, Z', V').
\end{aligned}
$$

  If $h' = h$, U_v accepts the signature and returns 1, otherwise rejects it and returns 0.
- **TS** U_v is able to generate a signature on a message m' which is indistinguishable from that created by U_a. In order to simulate a signature on a message m', U_v first selects two random values $l' \in_R \mathbb{Z}_q^*$ and $R' \in_R G_1$ and then computes:

$$
\begin{aligned}
&W' = l'P - Y_a, \quad K' = l's_v(D_a + q_a P_s),\\
&U' = H_3(m', K'),\\
&V' = x_v(R' + U' + W'), \quad Z' = r'_y x_v Y_a,\\
&h' = H_2(U', R', Z', V').
\end{aligned}
$$

  Then returns $\sigma' = (l', R', h')$.

5.2 On delegatability of Lin's scheme

Lin claimed that his scheme is non-delegatable and has tried to prove this claim in Theorem 1 of [17]. However, we show that his scheme is delegatable, as everyone who knows the common secret key of U_a and U_v, i.e., $Comm.Key = (s_a s_v P, x_a x_v P)$, is able to produce a valid signature σ^* on a message m by selecting random values $l^*, r^* \in_R \mathbb{Z}_q^*$ and computing:

$$
\begin{aligned}
&K^* = l^* s_a s_v P, \quad U^* = H_3(m, K^*),\\
&R^* = r^* P - U^* = (r_x^*, r_y^*)\\
&V^* = (r^* + l^*)Y_v - x_a x_v P, \quad Z^* = r_y^* x_a x_v P,\\
&h^* = H_2(U^*, R^*, Z^*, V^*).
\end{aligned}
$$

It is easy to check that $\sigma^* = (l^*, R^*, h^*)$ is a valid signature since it passes the DVer algorithm successfully as:

$$
\begin{aligned}
&W = l^* P - Y_a,\\
&K' = l^* s_v(D_a + q_a P_s) = l^* s_v s_a P = l^* s_a s_v P = K^*,\\
&U' = H_3(m, K') = H_3(m, K^*) = U^*,\\
&V' = x_v(R^* + U' + W) = x_v(r^* P - U^* + U^* + l^* P - Y_a)\\
&\quad = r^* x_v P + l^* x_v P - x_v x_a P = (r^* + l^*)Y_v - x_a x_v P = V^*\\
&Z' = r_y^* x_v Y_a = r_y^* x_v x_a P = r_y^* x_a x_v P = Z^*,\\
&h' = H_2(U', R^*, Z', V') = H_2(U^*, R^*, Z^*, V^*) = h^*.
\end{aligned}
$$

In other words, it is not necessary to know $SK_a = (x_a, s_a)$ (or $SK_v = (x_v, s_v)$) to create a valid signature and the knowl-

edge of only $Comm.Key = (s_a s_v P, x_a x_v P)$ is sufficient to generate a valid signature. Note that the signer U_a can compute $Comm.Key = (s_a(D_v + q_v P), x_a Y_v)$ and delegate the signing rights to every third party by only revealing $Comm.Key$ to him/her (and of course without revealing SK_a). As a result, in contrast to his claim, Lin's scheme is not located in the class of non-delegatable CL-DVS schemes.

5.3 On unforgeability of Lin's scheme

Lin also claimed that his scheme is unforgeable against $\mathcal{A}_I$ and $\mathcal{A}_{II}$ in the ROM based on CDH assumption and has tried to prove this claim in Theorem 4 of his paper [17]. However, there is an important drawback in the process of the proof of Theorem 4 which makes it unreliable. The drawback is that in the security proof against $\mathcal{A}_I$, the public keys of U_{a^*} and U_{v^*} are assigned as $PK_{a^*} = (cP, q_{a^*}, D_{a^*})$ and $PK_{v^*} = (dP, q_{v^*}, D_{v^*})$, respectively, where cP and dP are the inputs of the CDH problem which $\mathcal{B}$ is trying to solve it. If these public keys be replaced, then the claims in the following of the proof are no longer hold. However, as discussed in Sect. 4.2.1, the condition of not replacing PK_{a^*} (i.e., C3) is not reasonable at all. As a result, the unforgeability against $\mathcal{A}_I$ becomes questionable.

6 Our proposed scheme

6.1 The algorithms of our proposal

In this subsection, we propose our CL-DVS scheme based on the ideas of the certificateless signature scheme in [28] which is based on Waters' identity-based encryption scheme [29]. Generality, identities can be considered of arbitrary lengths and a hash function $H_u : \{0,1\}^* \longrightarrow \{0,1\}^{n_u}$ can be used to convert them to the specific length n_u. The algorithms of our scheme are as follows:

- **Setup** On input a security parameter λ, the KGC selects two multiplicative cyclic groups G_1 and G_2 of a large prime order p, a random generator g of G_1 and a bilinear pairing $e : G_1 \times G_1 \longrightarrow G_2$. It also selects a random $s \in_R \mathbb{Z}_p^*$ and sets $g_1 = g^s$. Furthermore, it picks random values $g_2, u' \in_R G_1$ and a random vector $\overrightarrow{u} = (u_i) \in_R G_1^{n_u}$, where $u_i \in_R G_1$, for $i = 1, \ldots, n_u$, then computes $T = e(g_1, g_2)$ and $T' = e(g_1, g)$. It also chooses two collision-resistant hash functions $H_u : \{0,1\}^* \longrightarrow \{0,1\}^{n_u}$ and $H : \{0,1\}^* \times G_1^8 \longrightarrow \mathbb{Z}_p^*$. The public system parameters are $params = \{G_1, G_2, p, e, g, g_1, g_2, u', \overrightarrow{u}, T, T', H_u, H\}$, and the master secret key is $msk = g^{s^2}$.
- **PPKE** On input an identity ID_U, the KGC calculates $H_u(ID_U)$. Let $u[i]$ denotes the ith bit of $H_u(ID_U)$ and $\mathcal{U}_U = \{i | u[i] = 1, i = 1, \ldots, n_u\}$. The KGC randomly selects $r_U \in_R \mathbb{Z}_p^*$ and computes d_U, as follows:

$$d_U = (d_{U,1}, d_{U,2}) = \left(g^{s^2} \left(u' \prod_{i \in \mathcal{U}_U} u_i \right)^{r_U}, g^{r_U} \right).$$

- **SSV** The user with identity ID_U selects a random $x_U \in_R \mathbb{Z}_p^*$ as his secret value.
- **SPrK** The user with identity ID_U picks a random $r'_U \in_R \mathbb{Z}_p^*$ and computes his private key as follows:

$$\begin{aligned} SK_U &= (SK_{U,1}, SK_{U,2}) \\ &= \left(d_{U,1}^{x_U^2} \left(u' \prod_{i \in \mathcal{U}_U} u_i \right)^{r'_U}, d_{U,2}^{x_U^2} g^{r'_U} \right) \\ &= \left(g^{s^2 x_U^2} \left(u' \prod_{i \in \mathcal{U}_U} u_i \right)^{r_U x_U^2 + r'_U}, g^{r_U x_U^2 + r'_U} \right) \end{aligned}$$

- **SPuK** The user with identity ID_U computes his public key as follows:

$$PK_U = (PK_{U,1}, PK_{U,2}, PK_{U,3}) = \left(g_1^{x_U}, g_2^{\frac{1}{x_U}}, g^{\frac{1}{x_U}} \right).$$

Then the user publishes PK_U without requiring it to be certified.

- **DSign** Suppose that a signer with identity ID_S, wants to create a signature on a message m for a designated verifier with identity ID_V. The signer selects a random $r_m \in_R \mathbb{Z}_p^*$ and executes the following steps:
  1. Checks whether $e(PK_{V,1}, PK_{V,2}) = T$ and $e(PK_{V,1}, PK_{V,3}) = T'$ hold or not. If one of the equalities does not hold, aborts and outputs $\perp$.
  2. Sets $\sigma_1 = SK_{S,2}$.
  3. Sets $\sigma_2 = g^{r_m}$.
  4. Computes $h = H(m, ID_S, ID_V, PK_S, PK_V, \sigma_1, \sigma_2)$.
  5. Sets $\sigma_3 = e(SK_{S,1} PK_{S,2}^{h r_m}, PK_{V,3})$.
  6. Outputs $\sigma = (\sigma_1, \sigma_2, \sigma_3)$.
- **DVer** Upon receiving (m, σ) from the signer, the designated verifier checks the validity of $\sigma = (\sigma_1, \sigma_2, \sigma_3)$ as follows:
  1. Checks whether $e(PK_{S,1}, PK_{S,2}) = T$ and $e(PK_{S,1}, PK_{S,3}) = T'$ hold or not. If one of the equalities does not hold, aborts and outputs $\perp$.

2. Computes $h = H(m, ID_S, ID_V, PK_S, PK_V, \sigma_1, \sigma_2)$.
3. Verifies the following equality:

$$\sigma_3 = (e(PK_{S,1}, PK_{S,1}).e\left(u' \prod_{i \in \mathcal{U}_S} u_i, \sigma_1\right) .e(PK_{S,2}^h, \sigma_2))^{\frac{1}{x_V}}. \quad (1)$$

If Eq. (1) holds, the verifier accepts the signature and outputs 1, otherwise rejects it and outputs 0.

– **TS** The verifier is able to generate a signature on m which is indistinguishable from that created by the real signer as follows:

1. Sets $\sigma_1 = SK_{S,2}$. Note that the verifier can obtain $SK_{S,2}$ from the previously signed messages by the signer.
2. Selects a random $r_m \in_R \mathbb{Z}_p^*$ and computes $\sigma_2 = g^{r_m}$.
3. Computes $h = H(m, ID_S, ID_V, PK_S, PK_V, \sigma_1, \sigma_2)$.
4. Computes:

$$\sigma_3 = (e(PK_{S,1}, PK_{S,1}).e\left(u' \prod_{i \in \mathcal{U}_S} u_i, \sigma_1\right) .e(PK_{S,2}^h, \sigma_2))^{\frac{1}{x_V}}.$$

Then returns $\sigma = (\sigma_1, \sigma_2, \sigma_3)$.

6.2 Some considerations

Note that the DVer algorithm is defined similarly in all proposed CL-DVS schemes in [9–17,19] as it takes $params$, a message/DVS pair (m, σ), ID_S, PK_S, ID_V and SK_V as input and outputs 1 if σ is valid and 0, otherwise. However, it is necessary for a strong DVS (not DVS) that DVer takes SK_V as input. In our proposed scheme (which is a DVS), the verifier can execute the DVer algorithm with only his secret value x_V, i.e., SK_V is not required to be an input of the DVer algorithm.

We consider the set of oracles that can be accessed by $\mathcal{A}_I$ and $\mathcal{A}_{II}$ as $\mathcal{O}_I = \{\mathcal{O}_d, \mathcal{O}_{PK}, \mathcal{O}_{SK}, \mathcal{O}_{RPK}, \mathcal{O}_S, \mathcal{O}_V\}$ and $\mathcal{O}_{II} = \{\mathcal{O}_{PK}, \mathcal{O}_{SK}, \mathcal{O}_S, \mathcal{O}_V\}$, respectively. As our proof is in the standard model, we do not consider $\mathcal{O}_H$, and as our scheme is based on SPrK2, we only consider $\mathcal{O}_{SK}$ without requiring to simulate $\mathcal{O}_x$ (See Sect. 4.1).

We say that $\mathcal{A}_I$ wins Game 1 if σ^* is a valid signature on m^* with respect to ID_{S^*} and ID_{V^*} and meanwhile σ^* is not obtained from the $\mathcal{O}_{DSign}$ oracle. Furthermore:

– ID_{S^*} has not been submitted to $\mathcal{O}_d$ and $\mathcal{O}_{SK}$.

– ID_{V^*} has not been submitted to $\mathcal{O}_{RPK}$.

The first condition is considered to prevent $\mathcal{A}_I$ from obtaining SK_{S^*}. Note that as only x_{V^*} (not SK_{V^*}) is used in the TS algorithm, there is not any reason to prevent $\mathcal{A}_I$ from obtaining SK_{V^*}. Moreover, the second case is C4 which is a reasonable condition as discussed in Sect. 4.2.1.

We say that $\mathcal{A}_{II}$ wins Game 2 if σ^* is a valid signature on m^* with respect to ID_{S^*} and ID_{V^*} and meanwhile σ^* is not obtained from the $\mathcal{O}_{DSign}$ oracle. Furthermore:

– ID_{S^*} has not been submitted to $\mathcal{O}_{SK}$.

This condition is considered to prevent $\mathcal{A}_{II}$ from obtaining SK_{S^*}.

According to the above descriptions, we use the following definition for proving the unforgeability of our scheme in Sect. 7.2:

Definition 1 A CL-DVS scheme is $(\varepsilon, t, q_d, q_{sk}, q_{pk}, q_r, q_s, q_v)$-EUF-CM&IDA if no adversaries ($\mathcal{A}_I$ and $\mathcal{A}_{II}$) running in time at most t, making at most q_d PPKE queries from $\mathcal{O}_d$ ($q_d = 0$ for $\mathcal{A}_{II}$), q_{sk} SPrK queries from $\mathcal{O}_{SK}$, q_{pk} SPuK queries from $\mathcal{O}_{PK}$, q_r Replace-Public-Key queries from $\mathcal{O}_{RPK}$ ($q_r = 0$ for $\mathcal{A}_{II}$), q_s DSign queries from $\mathcal{O}_S$ and q_v DVer queries from $\mathcal{O}_V$, wins Game 1 and Game 2 with probability at least ε.

7 Security analysis of the proposed scheme

7.1 Correctness

The correctness of the proposed scheme can be easily verified as:

$$\begin{aligned}
\sigma_3 &= e(SK_{S,1} PK_{S,2}^{hr_m}, PK_{V,3}) \\
&= e\left(g^{s^2 x_S^2} \left(u' \prod_{i \in \mathcal{U}_S} u_i\right)^{r_S x_S^2 + r'_S} PK_{S,2}^{hr_m}, g^{\frac{1}{x_V}}\right) \\
&= \left(e(g^{s^2 x_S^2}, g).e\left(\left(u' \prod_{i \in \mathcal{U}_S} u_i\right)^{r_S x_S^2 + r'_S}, g\right).e(PK_{S,2}^{hr_m}, g)\right)^{\frac{1}{x_V}} \\
&= \left(e(g_1^{x_S}, g_1^{x_S}).e\left(u' \prod_{i \in \mathcal{U}_S} u_i, g^{r_S x_S^2 + r'_S}\right).e(PK_{S,2}^h, g^{r_m})\right)^{\frac{1}{x_V}} \\
&= \left(e(PK_{S,1}, PK_{S,1}).e\left(u' \prod_{i \in \mathcal{U}_S} u_i, \sigma_1\right).e(PK_{S,2}^h, \sigma_2)\right)^{\frac{1}{x_V}},
\end{aligned}$$

which shows the correctness of Eq. (1).

7.2 Unforgeability

Theorem 1 *The proposed scheme is* $(\varepsilon, t, q_d, q_{sk}, q_{pk}, q_r, q_s, q_v)$*-unforgeable against* $\mathcal{A}_I$*, if the* (ε', t')*-*$(\mathcal{S}_2, 5)$*-CBDHE-Set assumption holds in* (G_1, G_2)*, where*

$$\varepsilon' \geq \frac{\varepsilon}{4(q_d + q_{sk} + q_s + q_v + 1)(n_u + 1)},$$
$$t' \leq t + order(((q_d + q_{sk} + q_s + q_v)n_u)T_M + (q_{pk} + q_d + q_{sk} + q_s + q_v)T_E + (q_r + q_s + q_v)T_P),$$

in which T_M*,* T_E *and* T_P *are the time for a multiplication and exponentiation in* G_1 *and a pairing computation, respectively.*

Proof Suppose that there exists a $(\varepsilon, t, q_d, q_{sk}, q_{pk}, q_r, q_s, q_v)$-type I adversary $\mathcal{A}_I$, who can break the EUF-CM&IDA in the proposed scheme according to Game 1 and forge a DVS σ^* on message m^* with respect to ID_{S^*} and ID_{V^*}. By this assumption, we can construct a simulator $\mathcal{B}$ that can use $\mathcal{A}_I$ as a subroutine and solve an instance of a $(\mathcal{S}_2, 5)$-CBDHE-Set problem with a probability at least ε' and in time at most t', which contradicts the (ε', t')-$(\mathcal{S}_2, 5)$-CBDHE-Set assumption in (G_1, G_2).
Consider two multiplicative cyclic groups G_1 and G_2 of a large prime order p and a random generator g of G_1. Let $e : G_1 \times G_1 \longrightarrow G_2$ be a bilinear pairing. Suppose that $\mathcal{B}$ is given a random $(\mathcal{S}_2, 5)$-CBDHE-Set challenge $(g \in G_1, A = g^a \in G_1, B = g^{a^2} \in G_1)$ and is requested to output $e(g, g)^{a^5} \in G_2$. In order to use $\mathcal{A}_I$ as a subroutine, $\mathcal{B}$ must simulate the challenger $\mathcal{C}$ and answer all $\mathcal{A}_I$'s queries in Game 1, i.e., the queries from the oracles in set $\mathcal{O}_I = \{\mathcal{O}_d, \mathcal{O}_{PK}, \mathcal{O}_{SK}, \mathcal{O}_{RPK}, \mathcal{O}_S, \mathcal{O}_V\}$. In order to respond these queries consistently, $\mathcal{B}$ generates a list $\mathcal{L} = \{(ID_U, d_U, x_U, PK_U, SK_U, sta_U = 0)\}$ which is initially empty. Then $\mathcal{B}$ plays Game 1 with $\mathcal{A}_I$ as follows:

Setup Let $l = 2(q_d + q_{sk} + q_s + q_v + 1)$ and assume that $l(n_u + 1) < p$. $\mathcal{B}$ chooses $k \in_R \{0, 1, \ldots, n_u\}$ (Note that $0 \leq kl < p$, as $l(n_u + 1) < p$). $\mathcal{B}$ also picks random values $x', x_1, \ldots, x_{n_u} \in_R \mathbb{Z}_l$, $y', y_1, \ldots, y_{n_u} \in_R \mathbb{Z}_p$, and $z \in_R \mathbb{Z}_p$. These values are kept internal to $\mathcal{B}$. Then $\mathcal{B}$ assigns a set of public parameters as follows:

$$g_1 = B = g^{a^2}, \qquad g_2 = g^z,$$
$$u' = B^{-kl+x'} g^{y'}, \qquad u_i = B^{x_i} g^{y_i} \quad (\text{for } \ i = 1, 2, \ldots, n_u).$$

$\mathcal{B}$ also computes $T = e(g_1, g_2)$ and $T' = e(g_1, g)$, then selects two collision-resistant hash functions $H_u : \{0,1\}^* \longrightarrow \{0,1\}^{n_u}$ and $H : \{0,1\}^* \times G_1^8 \longrightarrow \mathbb{Z}_p^*$ and gives $params = \{G_1, G_2, p, e, g, g_1, g_2, u', \overrightarrow{u}, T, T', H_u, H\}$ to $\mathcal{A}_I$. For simplicity of analysis, define two functions:

$$F(U) = x' + \sum_{i \in \mathcal{U}_U} x_i - kl \quad \text{and} \quad J(U) = y' + \sum_{i \in \mathcal{U}_U} y_i,$$

where $\mathcal{U}_U$ is defined similar to that in the proposed scheme. Then, we have

$$u' \prod_{i \in \mathcal{U}_U} u_i = B^{F(U)} g^{J(U)}.$$

Note that by these assignments, $\mathcal{B}$ does not know the master secret key, $msk = g^{a^4}$, and he/she must simulate $\mathcal{C}$ and answer all $\mathcal{A}_I$'s queries in Game 1 without the knowledge of msk. Moreover, note that by the mentioned settings, all distributions are identical to those in the real world from the perspective of $\mathcal{A}_I$.

Queries In this step, $\mathcal{A}_I$ sends polynomially bounded number of queries to the oracles in set $\mathcal{O}_I = \{\mathcal{O}_d, \mathcal{O}_{PK}, \mathcal{O}_{SK}, \mathcal{O}_{RPK}, \mathcal{O}_S, \mathcal{O}_V\}$. $\mathcal{B}$ responds to $\mathcal{A}_I$'s queries by simulating these oracles as follows:
$\mathcal{O}_d$. As $\mathcal{A}_I$ issues a partial private key query for an identity ID_U to $\mathcal{O}_d$, $\mathcal{B}$ looks up $\mathcal{L}$ to find d_U and sends it to $\mathcal{A}_I$. If d_U does not exist in $\mathcal{L}$, $\mathcal{B}$ tries to produce d_U without the knowledge of msk key as follows:

- If $F(U) = 0 \bmod p$, $\mathcal{B}$ aborts the simulation.
- If $F(U) \neq 0 \bmod p$, $\mathcal{B}$ picks a random $r_U \in \mathbb{Z}_p^*$ and constructs d_U as follows:

$$d_U = \left(B^{-\frac{J(U)}{F(U)}} \left(u' \prod_{i \in \mathcal{U}_U} u_i \right)^{r_U}, B^{-\frac{1}{F(U)}} g^{r_U} \right)$$
$$= \left(g^{a^4} \left(u' \prod_{i \in \mathcal{U}_U} u_i \right)^{\tilde{r}_U}, g^{\tilde{r}_U} \right) = (d_{U,1}, d_{U,2}),$$

where $\tilde{r}_U = r_U - \frac{a^2}{F(U)}$. Then $\mathcal{B}$ sends d_U to $\mathcal{A}_I$ and adds it in $\mathcal{L}$.

$\mathcal{O}_{PK}$. As $\mathcal{A}_I$ issues a public key query for an identity ID_U to $\mathcal{O}_{PK}$, $\mathcal{B}$ looks up $\mathcal{L}$ to find PK_U and sends it to $\mathcal{A}_I$. If PK_U does not exist in $\mathcal{L}$, $\mathcal{B}$ picks a random $x_U \in_R \mathbb{Z}_p^*$ and acts as follows:

- If $F(U) = 0 \bmod p$, $\mathcal{B}$ sets $PK_U = (g_1^{x_U}, g_2^{\frac{1}{x_U}}, g^{\frac{1}{x_U}})$. Note that by this assignment, the real secret value of ID_U is x_U, which is known to $\mathcal{B}$.
- If $F(U) \neq 0 \bmod p$, $\mathcal{B}$ sets $PK_U = (A^{\frac{1}{x_U}}, A^{zx_U}, A^{x_U})$. Note that by this assignment, the real secret value of ID_U is $\frac{1}{ax_U}$, which is unknown to $\mathcal{B}$.

Then, $\mathcal{B}$ sends PK_U to $\mathcal{A}_I$. Furthermore, $\mathcal{B}$ adds PK_U and its corresponding x_U in $\mathcal{L}$.

$\mathcal{O}_{RPK}$. Suppose that $\mathcal{A}_I$ requests to replace the public key of an identity ID_U, i.e., PK_U with respect to x_U with a new public key $PK'_U = (PK'_{U,1}, PK'_{U,2}, PK'_{U,3})$ with respect to x'_U. $\mathcal{B}$ firstly checks whether $e(PK'_{U,1}, PK'_{U,2}) = T$ and $e(PK'_{U,1}, PK'_{U,3}) = T'$ hold or not. If both equalities hold, $\mathcal{B}$ looks up $\mathcal{L}$ to replace (x_U, PK_U) with (x'_U, PK'_U) and sets $sta_U = 1$. If (x_U, PK_U) does not exist in $\mathcal{L}$, $\mathcal{B}$ directly sets $(x_U, PK_U) = (x'_U, PK'_U)$ and $sta_U = 1$ in $\mathcal{L}$. Note that $\mathcal{A}_I$ can only replace PK_U with a new correctly constructed $PK'_U = (g_1^{x'_U}, g_2^{\frac{1}{x'_U}}, g^{\frac{1}{x'_U}})$ as $e(PK'_{U,1}, PK'_{U,2}) = T$ and $e(PK'_{U,1}, PK'_{U,3}) = T'$ must be satisfied. So $\mathcal{A}_I$ knows x'_U corresponding to PK'_U.

$\mathcal{O}_{SK}$. As $\mathcal{A}_I$ issues a private key query for an identity ID_U to $\mathcal{O}_{SK}$, $\mathcal{B}$ looks up $\mathcal{L}$ to find SK_U and sends it to $\mathcal{A}_I$. If SK_U does not exist in $\mathcal{L}$, $\mathcal{B}$ acts as follows:

- If $F(U) = 0 \bmod p$, $\mathcal{B}$ aborts the simulation.
- If $F(U) \neq 0 \bmod p$, $\mathcal{B}$ acts as follows:
  - If $sta_U = 0$ (i.e., PK_U has not been replaced and hence the real secret value of ID_U is $\frac{1}{ax_U}$), $\mathcal{B}$ picks a random $r''_U \in \mathbb{Z}_p^*$ and sets:

$$\begin{aligned} SK_U &= \left(B^{\frac{1}{x_U^2}} \left(u' \prod_{i \in \mathcal{U}_U} u_i \right)^{r''_U}, g^{r''_U} \right) \\ &= \left(g^{a^4 \left(\frac{1}{ax_U}\right)^2} \left(u' \prod_{i \in \mathcal{U}_U} u_i \right)^{r''_U}, g^{r''_U} \right) \\ &= (SK_{U,1}, SK_{U,2}), \end{aligned}$$

    where $r''_U = \tilde{r}_U (\frac{1}{ax_U})^2 + r'_U$. Then, $\mathcal{B}$ returns SK_U to $\mathcal{A}_I$ and also stores it in $\mathcal{L}$.
  - If $sta_U = 1$ (i.e., PK_U has been replaced with PK'_U), $\mathcal{B}$ can retrieve x_U (which is in fact x'_U corresponding to PK'_U) from $\mathcal{L}$. Then $\mathcal{B}$ picks d_U from $\mathcal{L}$ if exists, otherwise $\mathcal{B}$ obtain d_U by simulating $\mathcal{O}_d$ as described (note that as $F(U) \neq 0 \bmod p$, $\mathcal{B}$ can simulate $\mathcal{O}_d$ without aborting). Finally, $\mathcal{B}$ can obtain SK_U by running the SPrK algorithm, since he knows both x_U and d_U. So $\mathcal{B}$ generates SK_U, returns it to $\mathcal{A}_I$ and also stores it in $\mathcal{L}$.

$\mathcal{O}_S$. As $\mathcal{A}_I$ issues a signature query for (m, ID_S, ID_V) to $\mathcal{O}_S$, $\mathcal{B}$ searches $\mathcal{L}$ to find SK_S. If this entry exists in $\mathcal{L}$, $\mathcal{B}$ picks it and creates a signature on m by running the DSign algorithm and returns it to $\mathcal{A}_I$, otherwise $\mathcal{B}$ acts as follows:

- If $F(S) \neq 0 \bmod p$, $\mathcal{B}$ obtains SK_S by simulating $\mathcal{O}_{SK}$, then creates a signature on m by running the DSign algorithm and returns it to $\mathcal{A}_I$.
- If $F(S) = 0 \bmod p$, $\mathcal{B}$ acts as follows:
  - If $sta_V = 1$ (i.e., PK_V has been replaced with PK'_V), $\mathcal{B}$ can retrieve x_V (which is in fact x'_V corresponding to PK'_V) from $\mathcal{L}$, then creates a signature on m by running the TS algorithm and returns it to $\mathcal{A}_I$.
  - If $sta_V = 0$ (i.e., PK_V has not been replaced), $\mathcal{B}$ acts as follows:
    - If $F(V)=0 \bmod p$, i.e., $PK_V=(g_1^{x_V}, g_2^{\frac{1}{x_V}}, g^{\frac{1}{x_V}})$, $\mathcal{B}$ can retrieve x_V from $\mathcal{L}$, then creates a signature on m by running the TS algorithm and returns it to $\mathcal{A}_I$.
    - If $F(V) \neq 0 \bmod p$, $\mathcal{B}$ aborts the simulation. (Note that in this case, $PK_V=(A^{\frac{1}{x_V}}, A^{zx_V}, A^{x_V})$, and $\mathcal{B}$ cannot retrieve the real secret value of ID_V, i.e., $\frac{1}{ax_V}$.)

$\mathcal{O}_V$. As $\mathcal{A}_I$ issues a verification query for $((m, \sigma = (\sigma_1, \sigma_2, \sigma_3)), ID_S, ID_V)$ to $\mathcal{O}_V$, $\mathcal{B}$ acts as follows:

- If $sta_V = 1$ (i.e., PK_V has been replaced with PK'_V), $\mathcal{B}$ can retrieve x_V (which is in fact x'_V corresponding to PK'_V) from $\mathcal{L}$, then verifies σ by running the DVer algorithm and sends the result to $\mathcal{A}_I$.
- If $sta_V = 0$ (i.e., PK_V has not been replaced), $\mathcal{B}$ acts as follows:
  - If $F(V) = 0 \bmod p$, i.e., $PK_V = (g_1^{x_V}, g_2^{\frac{1}{x_V}}, g^{\frac{1}{x_V}})$, $\mathcal{B}$ can retrieve x_V from $\mathcal{L}$, then verifies σ by running the DVer algorithm and sends the result to $\mathcal{A}_I$.
  - If $F(V) \neq 0 \bmod p$, $\mathcal{B}$ cannot retrieve the real secret value of ID_V, i.e., $\frac{1}{ax_V}$. In this case, $\mathcal{B}$ picks x_V. Then, $\mathcal{B}$ acts as follows:
    - If $F(S) \neq 0 \bmod p$, $\mathcal{B}$ obtains SK_S by simulating $\mathcal{O}_{SK}$. Then, $\mathcal{B}$ acts as follows:
      - If $sta_S = 0$, (i.e., $PK_S = (A^{\frac{1}{x_S}}, A^{zx_S}, A^{x_S})$ which has not been replaced), $\mathcal{B}$ retrieves x_S, computes $h = H(m, ID_S, ID_V, PK_S, PK_V, \sigma_1, \sigma_2)$ and verifies σ by checking the following equation:

$$\sigma_3 = e(SK_{S,1}, PK_{V,3}) e(\sigma_2, B)^{zhx_S x_V}$$

      - If $sta_S = 1$, (i.e., PK_S has been replaced with $PK'_S = (g_1^{x'_S}, g_2^{\frac{1}{x'_S}}, g^{\frac{1}{x'_S}})$ and x_S has been replaced with x'_S), $\mathcal{B}$ retrieves x_S from $\mathcal{L}$, computes $h = H(m, ID_S, ID_V, PK_S, PK_V, \sigma_1, \sigma_2)$ and verifies σ by checking the following equation:

$$\sigma_3 = e(SK_{S,1}, PK_{V,3}) e(\sigma_2, A)^{\frac{zhx_V}{x_S}}.$$

- If $F(S) = 0 \bmod p$, $\mathcal{B}$ aborts the simulation.

Forgery After a polynomially bounded number of queries (if $\mathcal{B}$ does not abort), $\mathcal{A}_I$ outputs a new valid signature $\sigma^* = (\sigma_1^*, \sigma_2^*, \sigma_3^*)$ on message m^* with respect to ID_{S^*} and ID_{V^*}. At the end of running Game 1 between $\mathcal{A}_I$ and $\mathcal{B}$, $\mathcal{B}$ acts as follows:

- If $F(S^*) \neq 0 \bmod p$ or $F(V^*) = 0 \bmod p$, $\mathcal{B}$ aborts.
- If $F(S^*) = 0 \bmod p$ and $F(V^*) \neq 0 \bmod p$, (and the simulation does not fail in any steps), $\mathcal{B}$ retrieves x_{S^*} and x_{V^*} from $\mathcal{L}$ and computes $h = H(m^*, ID_{S^*}, ID_{V^*}, PK_{S^*}, PK_{V^*}, \sigma_1^*, \sigma_2^*)$. Then $\mathcal{B}$ calculates and outputs:

$$e(g,g)^{a^5} = \left(\frac{\sigma_3^*}{e((\sigma_1^*)^{J(S^*)} (\sigma_2^*)^{\frac{zh^*}{x_{S^*}}}, A^{x_{V^*}})} \right)^{\frac{1}{x_{S^*}^2 x_{V^*}}},$$

as the solution to the $(\mathcal{S}_2, 5)$-CBDHE-Set problem instance.

Time Analysis Noting the above descriptions, we can see that $\mathcal{B}$ needs a time $t' \leq t + order(((q_d + q_{sk} + q_s + q_v)n_u)T_M + (q_{pk} + q_d + q_{sk} + q_s + q_v)T_E + (q_r + q_s + q_v)T_P)$, for running the game.

Probability Analysis Here, we evaluate the success probability of $\mathcal{B}$ for solving the $(\mathcal{S}_2, 5)$-CBDHE-Set problem instance, i.e., ε'. $\mathcal{B}$ can complete the simulation without aborting if all of the following independent events happen:

- E_1: $F(U) \neq 0 \bmod p$ for all queries from $\mathcal{O}_d$ and $\mathcal{O}_{SK}$.
- E_2: Let $\mathrm{E}_{2,1}$, $\mathrm{E}_{2,2}$, and $\mathrm{E}_{2,3}$ be the events of $F(S) = 0 \bmod p$, $F(V) \neq 0 \bmod p$, and $sta_V = 0$, respectively, for all queries from $\mathcal{O}_S$ and $\mathcal{O}_V$. Define $\mathrm{E}_2 = \overline{\mathrm{E}_{2,1} \bigcap \mathrm{E}_{2,2} \bigcap \mathrm{E}_{2,3}}$.
- E_3: $F(S^*) = 0 \bmod p$ and $F(V^*) \neq 0 \bmod p$.

Note that $\Pr[\mathrm{E}_{2,3}] = \frac{1}{2}$. Moreover, it is easy to see that [24]

$$\Pr[F(U) = 0 \bmod p] = \frac{1}{l(n_u+1)}.$$

So we have

$$\Pr[\mathrm{E}_1] = \left(1 - \frac{1}{l(n_u+1)}\right)^{q_d+q_{sk}},$$

$$\Pr[\mathrm{E}_2] = \left(1 - \frac{1}{2}.\left(1 - \frac{1}{l(n_u+1)}\right).\frac{1}{l(n_u+1)}\right)^{q_s+q_v} \geq \left(1 - \frac{1}{l(n_u+1)}\right)^{q_s+q_v},$$

$$\Pr[\mathrm{E}_3] = \left(1 - \frac{1}{l(n_u+1)}\right).\frac{1}{l(n_u+1)}.$$

Hence,

$$\begin{aligned}
\Pr[\overline{abort}] &\geq \Pr[\mathrm{E}_1 \bigcap \mathrm{E}_2 \bigcap \mathrm{E}_3] = \Pr[\mathrm{E}_1].\Pr[\mathrm{E}_2].\Pr[\mathrm{E}_3] \\
&\geq \left(1 - \frac{1}{l(n_u+1)}\right)^{q_d+q_{sk}} . \left(1 - \frac{1}{l(n_u+1)}\right)^{q_s+q_v} \\
&\quad . \left(1 - \frac{1}{l(n_u+1)}\right).\frac{1}{l(n_u+1)} \\
&= \left(1 - \frac{1}{l(n_u+1)}\right)^{q_d+q_{sk}+q_s+q_v+1} . \frac{1}{l(n_u+1)} \\
&\geq \left(1 - \frac{q_d+q_{sk}+q_s+q_v+1}{l(n_u+1)}\right).\frac{1}{l(n_u+1)} \\
&\geq \left(1 - \frac{q_d+q_{sk}+q_s+q_v+1}{l}\right).\frac{1}{l(n_u+1)} \\
&= \frac{1}{4(q_d+q_{sk}+q_s+q_v+1)(n_u+1)}, \qquad (2)
\end{aligned}$$

where the rightmost equality is implied from $l = 2(q_d + q_{sk} + q_s + q_v + 1)$.
Noting Eq. (2) and that ε is the success probability of $\mathcal{A}_I$ in Game 1, we have

$$\varepsilon' \geq \varepsilon.\Pr[\overline{abort}] \geq \frac{\varepsilon}{4(q_d+q_{sk}+q_s+q_v+1)(n_u+1)}.$$

As the final result, if $\mathcal{A}_I$ can win Game 1 with a non-negligible probability ε, then $\mathcal{B}$ can solve an instance of the $(\mathcal{S}_2, 5)$-CBDHE-Set problem with a non-negligible probability ε' and this is a contradiction of the $(\mathcal{S}_2, 5)$-CBDHE-Set assumption in complexity theory. □

Theorem 2 *The proposed scheme is $(\varepsilon, t, q_{sk}, q_{pk}, q_s, q_v)$-unforgeable against $\mathcal{A}_{II}$, if the (ε', t')-$(\mathcal{S}_4, 9)$-CBDHE-Set assumption holds in (G_1, G_2), where*

$$\varepsilon' \geq \frac{\varepsilon}{4(q_{sk}+q_s+q_v+1)(n_u+1)},$$
$$t' \leq t + order(((q_{sk}+q_s+q_v)n_u)T_M + (q_{pk}+q_{sk}+q_s+q_v)T_E + (q_s+q_v)T_P),$$

Proof Suppose that there exists a $(\varepsilon, t, q_{sk}, q_{pk}, q_s, q_v)$-type II adversary $\mathcal{A}_{II}$, who can break the EUF-CM&IDA in the proposed scheme according to Game 2 and forge a DVS σ^* on message m^* with respect to ID_{S^*} and ID_{V^*}. By this assumption, we can construct a simulator $\mathcal{B}$ that can use $\mathcal{A}_{II}$ as a subroutine and solve an instance of a $(\mathcal{S}_4, 9)$-CBDHE-Set problem with a probability at least ε' and in time at most t', which contradicts the (ε', t')-$(\mathcal{S}_4, 9)$-CBDHE-Set assumption in (G_1, G_2).
Consider two multiplicative cyclic groups G_1 and G_2 of a large prime order p and a random generator $g = C = f^{a^2}$ of G_1. Let $e : G_1 \times G_1 \longrightarrow G_2$. Suppose that $\mathcal{B}$ is

given a random $(\mathcal{S}_4, 9)$-CBDHE-Set challenge $(A = f \in G_1, B = f^a \in G_1, C = f^{a^2} \in G_1, D = f^{a^3} \in G_1, E = f^{a^4} \in G_1)$ and is requested to output $e(f, f)^{a^9} \in G_2$. In order to use $\mathcal{A}_{II}$ as a subroutine, $\mathcal{B}$ must simulate the challenger $\mathcal{C}$ and answer all $\mathcal{A}_{II}$'s queries from $\mathcal{O}_{II} = \{\mathcal{O}_{PK}, \mathcal{O}_{SK}, \mathcal{O}_S, \mathcal{O}_V\}$. In order to respond these queries consistently, $\mathcal{B}$ generates a list $\mathcal{L} = \{(ID_U, x_U, PK_U, SK_U)\}$ which is initially empty. Then $\mathcal{B}$ plays Game 2 with $\mathcal{A}_{II}$ as follows:

Setup Let $l = 2(q_{sk}+q_s+q_v+1)$ and assume that $l(n_u+1) < p$. $\mathcal{B}$ selects a random $s \in \mathbb{Z}_p^*$ and sets $g_1 = g^s = C^s$. Then $\mathcal{B}$ selects the values $k \in_R \{0, 1, \ldots, n_u\}, x', x_1, \ldots, x_{n_u} \in_R \mathbb{Z}_l, y', y_1, \ldots, y_{n_u} \in_R \mathbb{Z}_p$, and $z \in_R \mathbb{Z}_p^*$, and assigns:

$$g = C = f^{a^2}, \qquad g_1 = g^s = C^s, \qquad g_2 = g^z$$
$$u' = B^{-kl+x'} C^{y'}, \qquad u_i = B^{x_i} C^{y_i} \quad (\text{for } \ i = 1, 2, \ldots, n_u).$$

$\mathcal{B}$ also computes $T = e(g_1, g_2)$ and $T' = e(g_1, g)$, then selects two collision-resistant hash functions $H_u : \{0,1\}^* \longrightarrow \{0,1\}^{n_u}$ and $H : \{0,1\}^* \times G_1^8 \longrightarrow \mathbb{Z}_p^*$ and gives $params = \{G_1, G_2, p, e, g, g_1, g_2, u', \overrightarrow{u}, T, T', H_u, H\}$ to $\mathcal{A}_{II}$. $\mathcal{B}$ also computes two functions $F(U)$ and $J(U)$ similar to those in the proof of Theorem 1 and sends them to $\mathcal{A}_{II}$. By these assignments, we have

$$u' \prod_{i \in \mathcal{U}_U} u_i = B^{F(U)} g^{J(U)}.$$

Note that by the mentioned settings in this step, all distributions are identical to those in the real world from the perspective of $\mathcal{A}_{II}$. Also, remember that in Game 2 (in contrast to Game 1), $\mathcal{B}$ knows the master secret key, $msk = g^{s^2}$ and must simulate $\mathcal{C}$ and answer all $\mathcal{A}_{II}$'s queries by this fact.

Queries In this step, $\mathcal{A}_{II}$ has access to the oracles in set $\mathcal{O}_{II} = \{\mathcal{O}_{PK}, \mathcal{O}_{SK}, \mathcal{O}_S, \mathcal{O}_V\}$. $\mathcal{B}$ responds to $\mathcal{A}_{II}$' s queries by simulating these oracles as follows:

$\mathcal{O}_{PK}$. As $\mathcal{A}_{II}$ sends a public key query for an identity ID_U to $\mathcal{O}_{PK}$, $\mathcal{B}$ checks whether such key exists in $\mathcal{L}$. If so, $\mathcal{B}$ returns this public key to $\mathcal{A}_{II}$. Otherwise, $\mathcal{B}$ picks a random $x_U \in_R \mathbb{Z}_p^*$ and acts as follows:

- If $F(U) = 0 \bmod p$, $\mathcal{B}$ sets $PK_U = (E^{sx_U}, A^{\frac{z}{x_U}}, A^{\frac{1}{x_U}})$. Note that by this assignment, the real secret value of ID_U is $a^2 x_U$, which is known to $\mathcal{B}$.
- If $F(U) \neq 0 \bmod p$, $\mathcal{B}$ sets $PK_U = (B^{\frac{s}{x_U}}, D^{zx_U}, D^{x_U})$. Note that by this assignment, the real secret value of ID_U is $\frac{1}{ax_U}$, which is unknown to $\mathcal{B}$.

Then $\mathcal{B}$ sends PK_U to $\mathcal{A}_{II}$. Furthermore, $\mathcal{B}$ adds PK_U and its corresponding x_U in $\mathcal{L}$.

$\mathcal{O}_{SK}$. As $\mathcal{A}_{II}$ sends a private key query for an identity ID_U to $\mathcal{O}_{SK}$, $\mathcal{B}$ checks whether such key exists in $\mathcal{L}$. If so, $\mathcal{B}$ returns this private key to $\mathcal{A}_{II}$. Otherwise $\mathcal{B}$ acts as follows:

- If $F(U) = 0 \bmod p$, $\mathcal{B}$ aborts the simulation.
- If $F(U) \neq 0 \bmod p$, $\mathcal{B}$ checks whether (x_U, Pk_U) exists in $\mathcal{L}$. If so, $\mathcal{B}$ picks it, otherwise $\mathcal{B}$ produces (x_U, Pk_U) by simulating $\mathcal{O}_{PK}$ (Note that in as $F(U) \neq 0 \bmod p$, the real secret value of ID_U is $\frac{1}{ax_U}$). Then $\mathcal{B}$ selects a random $r''_U \in_R \mathbb{Z}_p^*$ and assigns the private key as:

$$\begin{aligned} SK_U &= \left(B^{-\frac{J(U)}{F(U)}\left(\frac{s^2}{x_U^2}\right)} \left(u' \prod_{i \in \mathcal{U}_U} u_i \right)^{r''_U}, B^{-\frac{s^2}{x_U^2 F(U)}} g^{r''_U} \right) \\ &= \left(g^{s^2\left(\frac{1}{ax_U}\right)^2} \left(u' \prod_{i \in \mathcal{U}_U} u_i \right)^{\tilde{r}''_U}, g^{\tilde{r}''_U} \right) \\ &= (SK_{U,1}, SK_{U,2}), \end{aligned}$$

  where $\tilde{r}''_U = r''_U - \frac{s^2}{ax_U^2 F(U)}$. Then $\mathcal{B}$ returns SK_U to $\mathcal{A}_{II}$ and also adds it in $\mathcal{L}$.

$\mathcal{O}_S$. As $\mathcal{A}_{II}$ sends a signing query for (m, ID_S, ID_V), $\mathcal{B}$ searches $\mathcal{L}$ to find SK_S. If this entry exists, $\mathcal{B}$ picks it and creates a DVS σ on message m by running the DSign algorithm and sends it to $\mathcal{A}_{II}$. Otherwise, $\mathcal{B}$ acts as follows:

- If $F(S) = 0 \bmod p$, $\mathcal{B}$ aborts the simulation.
- If $F(S) \neq 0 \bmod p$, $\mathcal{B}$ obtains SK_S by simulating $\mathcal{O}_{SK}$, then creates a DVS σ on message m by running the DSign algorithm and sends it to $\mathcal{A}_{II}$.

$\mathcal{O}_V$. As $\mathcal{A}_{II}$ sends a verification query for $(\sigma = (\sigma_1, \sigma_2, \sigma_3), ID_S, ID_V)$, $\mathcal{B}$ acts as follows:

- If $F(S) = 0 \bmod p$, $\mathcal{B}$ aborts the simulation.
- If $F(S) \neq 0 \bmod p$ (and so $PK_S = (B^{\frac{s}{x_S}}, D^{zx_S}, D^{x_S})$, $\mathcal{B}$ obtains SK_S by simulating $\mathcal{O}_{SK}$. Then $\mathcal{B}$ computes $h = H(m, ID_S, ID_V, PK_S, PK_V, \sigma_1, \sigma_2)$, retrieves x_S from $\mathcal{L}$, and acts as follows:
  - If $F(V) = 0 \bmod p$, i.e., $PK_V = (E^{sx_V}, A^{\frac{z}{x_V}}, A^{\frac{1}{x_V}})$, $\mathcal{B}$ retrieves x_V from $\mathcal{L}$ and verifies σ by checking the following equation:

    $$\sigma_3 = e(SK_{S,1}, PK_{V,3}) e(\sigma_2, B)^{\frac{zhx_S}{x_V}}.$$

  - If $F(V) \neq 0 \bmod p$, i.e., $PK_V = (B^{\frac{s}{x_V}}, D^{zx_V}, D^{x_V})$, $\mathcal{B}$ $\mathcal{B}$ retrieves x_V from $\mathcal{L}$ and verifies σ by checking the following equation:

$$\sigma_3 = e(SK_{S,1}, PK_{V,3})e(\sigma_2, E)^{zhx_S x_V}.$$

Forgery After a polynomially bounded number of queries (if $\mathcal{B}$ does not abort), $\mathcal{A}_{II}$ outputs a new valid DVS $\sigma^* = (\sigma_1^*, \sigma_2^*, \sigma_3^*)$ on message m^* for ID_{S^*} and ID_{V^*}.
At the end of running Game 2 between $\mathcal{A}_{II}$ and $\mathcal{B}$, $\mathcal{B}$ acts as follows:

- If $F(S^*) \neq 0 \bmod p$ or $F(V^*) = 0 \bmod p$, $\mathcal{B}$ aborts.
- If $F(S^*) = 0 \bmod p$ and $F(V^*) \neq 0 \bmod p$, (and the simulation does not fail in any steps), $\mathcal{B}$ retrieves x_{S^*} and x_{V^*} from $\mathcal{L}$ and computes $h = H(m^*, ID_{S^*}, ID_{V^*}, PK_{S^*}, PK_{V^*}, \sigma_1^*, \sigma_2^*)$. Then $\mathcal{B}$ calculates and outputs:

$$e(f, f)^{a^9} = \left(\frac{\sigma_3^*}{e((\sigma_1^*)^{J(S^*)}, D^{x_{V^*}}) e(\sigma_2^*, B^{\frac{zh^* x_{V^*}}{x_{S^*}}})} \right)^{\frac{1}{s^2 x_{S^*}^2 x_{V^*}}},$$

as the solution to the $(\mathcal{S}_4, 9)$-CBDHE-Set problem instance.

Time Analysis Noting the above descriptions, we can see that $\mathcal{B}$ needs a time $t' \leq t + order(((q_{sk} + q_s + q_v)n_u)T_M + (q_{pk} + q_{sk} + q_s + q_v)T_E + (q_s + q_v)T_P)$, for running the game.
Probability Analysis Here, we evaluate the success probability of $\mathcal{B}$ for solving the $(\mathcal{S}_4, 9)$-CBDHE-Set problem instance, i.e., ε'. $\mathcal{B}$ can complete the simulation without aborting if all of the following independent events happen:

- $\mathbf{E}_1$: $F(U) \neq 0 \bmod p$ for all queries from $\mathcal{O}_{SK}$.
- $\mathbf{E}_2$: $F(S) \neq 0 \bmod p$ for all queries from $\mathcal{O}_S$ and $\mathcal{O}_V$.
- $\mathbf{E}_3$: $F(S^*) = 0 \bmod p$ and $F(V^*) \neq 0 \bmod p$.

Noting $\Pr[F(U) = 0 \bmod p] = \frac{1}{l(n_u+1)}$ [24], we have

$$\Pr[\mathrm{E}_1] = \left(1 - \frac{1}{l(n_u+1)}\right)^{q_{sk}},$$
$$\Pr[\mathrm{E}_2] = \left(1 - \frac{1}{l(n_u+1)}\right)^{q_s+q_v},$$
$$\Pr[\mathrm{E}_3] = \left(1 - \frac{1}{l(n_u+1)}\right) \cdot \frac{1}{l(n_u+1)}.$$

Hence,

$$\begin{aligned}
\Pr[\overline{abort}] &\geq \Pr[\mathrm{E}_1 \bigcap \mathrm{E}_2 \bigcap \mathrm{E}_3] = \Pr[\mathrm{E}_1].\Pr[\mathrm{E}_2].\Pr[\mathrm{E}_3] \\
&\geq \left(1 - \frac{1}{l(n_u+1)}\right)^{q_{sk}} \cdot \left(1 - \frac{1}{l(n_u+1)}\right)^{q_s+q_v} \\
&\quad \cdot \left(1 - \frac{1}{l(n_u+1)}\right) \cdot \frac{1}{l(n_u+1)} \\
&= \left(1 - \frac{1}{l(n_u+1)}\right)^{q_{sk}+q_s+q_v+1} \cdot \frac{1}{l(n_u+1)} \\
&\geq \left(1 - \frac{q_{sk}+q_s+q_v+1}{l(n_u+1)}\right) \cdot \frac{1}{l(n_u+1)} \\
&\geq \left(1 - \frac{q_{sk}+q_s+q_v+1}{l}\right) \cdot \frac{1}{l(n_u+1)} \\
&= \frac{1}{4(q_{sk}+q_s+q_v+1)(n_u+1)} \qquad (3)
\end{aligned}$$

where the rightmost equality is implied from $l = 2(q_{sk} + q_s + q_v + 1)$.
Noting Eq. (3) and that ε is the success probability of $\mathcal{A}_{II}$ in Game 2, we have:

$$\varepsilon' \geq \varepsilon.\Pr[\overline{abort}] \geq \frac{\varepsilon}{4(q_{sk}+q_s+q_v+1)(n_u+1)}.$$

As the final result, if $\mathcal{A}_{II}$ can win Game 2 with a non-negligible probability ε, then $\mathcal{B}$ can solve an instance of the $(\mathcal{S}_4, 9)$-CBDHE-Set problem with a non-negligible probability ε' and this is a contradiction of the $(\mathcal{S}_4, 9)$-CBDHE-Set assumption in complexity theory. □

Theorem 3 *The proposed scheme is EUF-CM&IDA (according to Definition 1) in the standard model under the* $(\mathcal{S}_4, 9)$*-CBDHE-Set assumption.*

Proof The proof is directly implied from Theorems 1 and 2. □

7.3 Non-transferability

Theorem 4 *The proposed CL-DVS scheme is unconditionally non-transferable.*

Proof Suppose that σ is a DVS on message m with respect to ID_S and ID_V which is created by the original signer (by the DSign algorithm) and σ' is a DVS on m with respect to ID_S and ID_V which is produced by the designated verifier (by the TS algorithm).

In order to generate σ, the signer, with the private key SK_S, selects a random value $r_0 \in_R \mathbb{Z}_p^*$ and sets:

$$(\sigma_1, \sigma_2, \sigma_3) = (SK_{S,2}, g^{r_0}, e(SK_{S,1} PK_{S,2}^{hr_0}, PK_{V,3})),$$

where $h = H(m, ID_S, ID_V, PK_S, PK_V, \sigma_1, \sigma_2)$.

In order to generate σ', the designated verifier, with the secret value x_V, picks a random value $r_1 \in_R \mathbb{Z}_p^*$ and sets:

$$(\sigma_1', \sigma_2', \sigma_3') = \Big(SK_{S,2}, g^{r_1}, (e(PK_{S,1}, PK_{S,1}) \cdot e\left(u' \prod_{i \in \mathcal{U}_S} u_i, \sigma_1'\right) \cdot e(PK_{S,2}^{h'}, \sigma_2'))^{\frac{1}{x_V}}\Big),$$

where $h' = H(m, ID_S, ID_V, PK_S, PK_V, \sigma_1', \sigma_2')$.

It is easy to see that σ and σ' have the same distributions and hence they are indistinguishable. Suppose that a challenger $\mathcal{C}$ selects a random value $r^* \in_R \mathbb{Z}_p^*$, picks a bit $b \in_R \{0, 1\}$ by flipping a fair coin and creates a signature $\sigma^* = (\sigma_1^*, \sigma_2^*, \sigma_3^*)$ as follows:

– If $b = 0$, $\mathcal{C}$ sets:

$$(\sigma_1^*, \sigma_2^*, \sigma_3^*) = (SK_{S,2}, g^{r^*}, e(SK_{S,1} PK_{S,2}^{h^* r^*}, PK_{V,3})),$$

– If $b = 1$, $\mathcal{C}$ sets:

$$(\sigma_1^*, \sigma_2^*, \sigma_3^*) = \Big(SK_{S,2}, g^{r^*}, (e(PK_{S,1}, PK_{S,1}) .e\left(u' \prod_{i \in \mathcal{U}_S} u_i, \sigma_1^*\right) .e\left(PK_{S,2}^{h^*}, \sigma_2^*\right)\Big)^{\frac{1}{x_V}}\Big),$$

where $h^* = H(m, ID_S, ID_V, PK_S, PK_V, \sigma_1^*, \sigma_2^*)$. We have

$$\Pr[\sigma^* = \sigma] = \Pr[r^* = r_0] = \frac{1}{p-1},$$
$$\Pr[\sigma^* = \sigma'] = \Pr[r^* = r_1] = \frac{1}{p-1}.$$

Therefore, the distributions of σ and σ' are identical and a distinguisher $\mathcal{D}$ cannot distinguish whether the signature is created by the signer or the designated verifier. Hence, the signature is unconditionally non-transferable. □

Remark 4 As mentioned in Remark 3, our proposal is not placed in the category of non-delegatable CL-DVS schemes, since the signer can delegate his/her signing rights to every third party by providing $Comm.Key = e(SK_{S,1}, PK_{V,3})$ to him/her. Note that everyone who knows $Comm.Key = e(SK_{S,1}, PK_{V,3})$ can produce a valid DVS by selecting a random value $r \in_R \mathbb{Z}_p^*$ and setting:

$$(\sigma_1, \sigma_2, \sigma_3) = (SK_{S,2}, g^r, e(SK_{S,1}, PK_{V,3}) e(PK_{S,2}^{hr}, PK_{V,3})),$$

where $h = H(m, ID_S, ID_V, PK_S, PK_V, \sigma_1, \sigma_2)$. Note that $SK_{S,2}$ can be easily obtained from the previously signed messages from S to V.

8 Comparison

To the best of our knowledge, only nine CL-DVS schemes have been proposed in the literature to date [9–17]. In Table 3, a comparison among the existing CL-DVS schemes with our

Table 3 Comparisons between CL-DVS schemes

Scheme	Signature size	Signing cost		Verification cost		Standard model?
		Off-line	Online	Off-line	Online	
[9]	$1\lvert\mathbb{Z}_p^*\rvert$	$1P + 1E_{G_1}$	$1H$	$3P + 1E_{G_1}$	$1H$	×
[10]	$1\lvert\mathbb{Z}_p^*\rvert$	$1P + 1E_{G_1}$	$1H$	$1P + 1E_{G_1}$	$1H$	×
[11]	$2\lvert\mathbb{Z}_p^*\rvert + 1\lvert G_1\rvert$	$1P + 1E_{G_1}$	$1E_{G_2} + 2E_{G_1} + 1H$	–	$2P + 3E_{G_1} + 1H$	×
[12]	$1\lvert G_1\rvert + 1\lvert G_2\rvert$	$1E_{G_1}$	$1P + 3E_{G_1} + 1H$	$1E_{G_1}$	$1P + 1E_{G_1} + 1H$	×
[13]	$2\lvert G_1\rvert$	$1E_{G_1}$	$2E_{G_1} + 1H$	$1P + 1E_{G_1}$	$1P + 1E_{G_2} + 1H$	×
[14]	$2\lvert G_2\rvert$	$2P + 1E_{G_1}$	$1P + 1E_{G_2} + 2E_{G_1} + 2H$	$1P$	$1E_{G_2} + 1H$	×
[15]	$3\lvert\mathbb{Z}_p^*\rvert$	$1H$	$1E_G + 1H$	$1E_G + 1H$	$2E_G + 1H$	×
[16]	$2\lvert\mathbb{Z}_p^*\rvert + 1\lvert G_1\rvert$	$1P$	$3E_{G_1} + 2H$	$1P$	$3E_{G_1} + 2H$	×
[17]	$2\lvert\mathbb{Z}_p^*\rvert + 1\lvert G\rvert$	$2E_G$	$4E_G + 2H$	$2E_G$	$4E_G + 2H$	×
Ours	$2\lvert G_1\rvert + 1\lvert G_2\rvert$	$4P$	$1E_{G_2} + 1E_{G_1} + 1H$	$3P$	$2P + 1E_{G_1} + 1E_{G_2} + 1H$	✓

proposal is provided. As mentioned in Sect. 4, among all CL-DVS schemes in [9–17], there are issues in their security proofs, as outlined below:

- In the proof of the scheme in [9], EUF-CM&IDA (which is well known in CL-PKC) is not considered. Furthermore, the scheme is vulnerable against $\mathcal{A}_{II}$ according to the proposed attack in [19].
- In the proof of the scheme in [10], EUF-CM&IDA is not considered.
- In the proof of the scheme in [11], oracles are not simulated, even EUF-CMA is not considered. Moreover, probability analysis is not provided.
- In the proof of the schemes in [12,13], probability analysis is not provided.
- In the proof of the scheme in [14], probability analysis is not provided. Moreover, the scheme is vulnerable against $\mathcal{A}_{II}$ and key-compromise attack according to the proposed attacks in [20,21].
- In the proof of the scheme in [15], the oracle $\mathcal{O}_V$ is not simulated.
- In the proof of the scheme in [16], probability analysis is not provided. Moreover, the scheme is vulnerable against $\mathcal{A}_I$ according to the proposed attack in [18].
- In the proof of the scheme in [17], the unreasonable condition C3 makes the proof unreliable (See Sect. 4.2.1).

As a result, our scheme is not only the first scheme with reliable security proofs, but also the only scheme in the standard model. We should note that the computational cost of our proposal increases due to the need to provide additional security properties in the standard model. Nevertheless, we also note that most of these computations can be computed off-line (i.e., can be pre-computed before running the DSign and DVer algorithms), as shown in Table 3.

9 Conclusion

In this paper, we provided a discussion on the proposed certificateless designated verifier signature (CL-DVS) schemes and showed that the security proofs of none of them are reliable. Furthermore, we concentrate on the recently proposed CL-SDVS scheme (IEEE Access 2018) and showed that this scheme is delegatable in contrast to its author's claim. Moreover, we showed a drawback in the security proof of this scheme which makes it unreliable. Finally, we proposed the first CL-DVS scheme in the standard model with reliable security proofs. We proved the unforgeability of our scheme against the key replacement attacker ($\mathcal{A}_I$) and the malicious KGC attacker ($\mathcal{A}_{II}$) based on $(\mathcal{S}_2, 5)$-CBDHE-Set and $(\mathcal{S}_4, 9)$-CBDHE-Set assumptions, respectively. The security proof of our proposal does not suffer from the drawbacks in the proofs of existing CL-DVS schemes.

Compliance with ethical standards

Conflict of interest Authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (1978)
2. Chaum, D., Antwerpen, H.V.: Undeniable signatures. In Advances in Cryptology, CRYPTO'89 Proceedings, pp. 212–216. Springer, New York (1989)
3. Jakobsson, M., Sako, K., Impagliazzo, R.: Designated verifier proofs and their applications. In Advances in Cryptology, EUROCRYPT'96, pp. 143–154. Springer, Berlin (1996)
4. Chaum, D.: Private signature and proof systems, U.S. Patent 5,493,614
5. Wang, D., Wang, P.: Two birds with one stone: two-factor authentication with security beyond conventional bound. IEEE Trans. Dependable Secure Comput. **15**(4), 708–722 (2018)
6. Wang, D., Wang, N., Wang, P., Qing, S.: Preserving privacy for free: efficient and provably secure two-factor authentication scheme with user anonymity. Inf. Sci. **321**, 162–178 (2015)
7. Shamir, A.: Identity-based cryptosystem and signature scheme. In: Advances in Cryptology, Crypto 84. Springer, LNCS, vol. 196, pp. 47–53 (1984)
8. Al-Riyami, S.S., Paterson, K.: Certificateless public key cryptography. In: Asiacrypt 2003, Springer, LNCS , vol. 2894, pp. 452–473 (2003)
9. Huang, X., Susilo, W., Mu, Y., Zhang F.: Certificateless designated verifier signature schemes. In: 20th International Conference on Advanced Information Networking and Applications (AINA'06), Vienna, Australia, pp. 15–19 (2006)
10. Chen, H., Song, R., Zhang, F., Song, F.: An efficient certificateless short designated verifier signature scheme. In: 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'08), Dalian, China, pp. 1–6 (2008)
11. Du, H., Wen, Q.: Efficient certificateless designated verifier signatures and proxy signatures. Chin. J. Electron. **18**(1), 95–100 (2009)
12. Yang, B., Hu, Z., Xiao, Z.: Efficient certificateless strong designated verifier signature scheme. In: International Conference on Computational Intelligence and Security (CIS'09), Beijing, China, vol. 1, pp. 432–436 (2009)
13. Xiao, Z., Yang, B., Li, S.: Certificateless strong designated verifier signature scheme. In: 2nd International Conference on e-Business and Information System Security (EBISS), pp. 1–5. IEEE (2010)
14. Islam, S.H., Biswas, G.P.: Provably secure certificateless strong designated verifier signature scheme based on elliptic curve bilinear pairings. J. King Saud Univ. Comput. Inf. Sci. **25**(1), 51–61 (2013)
15. He, D., Chen, J.: An efficient certificateless designated verifier signature scheme. Int. Arab J. Inf. Technol. **10**(4), 389–396 (2013)
16. Chen, Y., Zhao, Y., Xiong, H., Yue, F.: A certificateless strong designated verifier signature scheme with non-delegatability. IJ Netw. Secur. **19**(4), 573–582 (2017)

17. Lin, H.Y.: A new certificateless strong designated verifier signature scheme: non-delegatable and SSA-KCA secure. IEEE Access **6**, 50765–50775 (2018)
18. Pakniat, N.: On the security of a certificateless strong designated verifier signature scheme. IACR Cryptology ePrint Archive (2018)
19. Au, M.H., Mu, Y., Chen, J., Wong, D.S., Liu, J.K., Yang, G.: Malicious KGC attacks in certificateless cryptography. In: Proceedings of the 2nd ACM symposium on Information, computer and communications security, ACM, pp. 302–311 (2007)
20. Liu, T., Wang, X., Ding, X.: security analysis and improvement of certificateless strong designated verifier signature scheme. Comput. Sci. **40**(7), 126–128 (2013). **(in chinese)**
21. Lin, H.Y., Ting, P.Y., Yang, L.F.: On the security of a provably secure certificateless strong designated verifier signature scheme based on bilinear pairings. ICTCE, pp. 61–65 (2017)
22. Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: Proceedings of the 1st ACM Conference on Computer and Communications Security, ACM, pp. 62–73 (1993)
23. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. In: Advances in CryptologyCRYPTO 2001, Springer, Berlin, pp. 213–229 (2001)
24. Gentry, C., Halevi, S.: Hierarchical identity based encryption with polynomially many levels. In: Theory of Cryptography Conference, pp. 437–456. Springer, Berlin (2009)
25. Li, Y., Lipmaa, H., Pei, D.: On delegatability of four designated verifier signatures. ICICS **3783**, 61–71 (2005)
26. Tian, H., Jiang, Z., Liu, Y., Wei, B.: A non-delegatable strong designated verifier signature without random oracles. In: 4th International Conference on Intelligent Networking and Collaborative Systems (INCoS), IEEE, pp. 237–244 (2012)
27. Dent, A.W.: A survey of certificateless encryption schemes and security models. Int. J. Inf. Secur. **7**(5), 349–377 (2008)
28. Yuan, Y., Wang, C.: Certificateless signature scheme with security enhanced in the standard model. Inf. Process. Lett. **114**(9), 492–499 (2014)
29. Waters, B.: Efficient identity-based encryption without random oracles. In: Eurocrypt'05 3494 , pp. 114–127 (2005)

Parvin Rastegari received the B.Sc., M.Sc. and PhD degrees in electrical engineering from Department of Electrical and Computer Engineering, Isfahan University of Technology in 2008, 2011 and 2019 respectively. Her M.Sc. dissertation is in the field of information theory entitled "The redundancy of some source codes" and her PhD thesis is in the field of information security entitled "Privacy-preserving digital signatures". Her current interested research topics are digital signatures and advanced security protocols.

Willy Susilo received a Ph.D. degree in Computer Science from University of Wollongong, Australia. He is a Professor and the Head of School of Computing and Information Technology and the director of Institute of Cybersecurity and Cryptology (iC^2) at the University o f Wollongong. He was previously awarded the prestigious ARC Future Fellow by the Australian Research Council (ARC) and the Researcher of the Year award in 2016 by the University of Wollongong. His main research interests include cybersecurity, cryptography and information security. His work has been cited more than 13,000 times in Google Scholar. He is the Editor-in-Chief of the Information journal. He has served as a program committee member in dozens of international conferences. He is currently serving as an Associate Editors in several international journals. He has published more than 400 research papers in the area of cybersecurity and cryptology.

Mohammad Dakhilalian Received the B.S. and Ph.D. degrees in Electrical Engineering from Isfahan University of Technology (IUT) in 1989 and 1998, respectively, and M.S. degree in Electrical Engineering from Tarbiat Modares University in 1993. He was an Assistant Professor of Faculty of Information & Communication Technology, Ministry of ICT, Tehran, Iran in 1999–2001. He joined IUT in 2001 and is an associate professor in Electrical and Computer Engineering Department. His current research interests are cryptography and data security.