Constructing Strong Identity-Based Designated Verifier Signatures with Self-Unverifiability

JuHee Ki, Jung Yeon Hwang, DaeHun Nyang, Beom-Hwan Chang,
Dong Hoon Lee, and Jong-in Lim

An identity-based strong designated verifier signature scheme provides restricted verifiability only for a verifier designated by a signer and proper privacy for the signer.

In this paper, we show that strong designated verifier signature schemes do not satisfy the self-unverifiability requirement in the sense that not only exposure of the verifier's secret key but also of the signer's secret key enables an attacker to verify signatures, which should have been the exclusive right of the verifier. We also present a generic method to construct a strong identity-based designated verifier signature scheme with self-unverifiability from identity-based key encapsulation and identity-based key sharing schemes. We prove that a scheme constructed from our method achieves unforgeability, non-transferability, and self-unverifiability if the two underlying components are secure. To show the advantage of our method, we present an example that outputs short signatures and we analyze its performance.

Keywords: Identity-based designated verifier signature, privacy, strongness, self-unverifiability.

I. Introduction

Relaxing the non-repudiation property of a standard digital signature, Jakobsson and others [1] introduced a specific type of signature for signer ambiguity, called designated verifier signature (DVS). A DVS scheme allows only a designated verifier to confirm validity of a given signature. This limited verifiability can be achieved by a sharing of signing capability between a signer and a designated verifier; in other words, a signature can be generated by not only a signer but also a designated verifier. When a designated verifier receives a signature from a signer, if the verifier did not generate the signature, the verifier is able to confirm that the signature is originated from the signer. Though anybody can publicly verify the validity of a signature, one cannot confirm the exact generator of this signature because both the signer and the verifier have signing capability. Here, the validity means that a signature has been generated either by a signer or by a designated verifier. This security property is formally known as *non-transferability* or *simulatability* [2]. In this sense, the non-transferability property provides signer ambiguity. An identity-based extension of a DVS scheme, say identity-based DVS (IBDVS) scheme, has been proposed to enjoy the benefit that an arbitrary public string such as an email address or a phone number may be used as a user's public key instead of requiring public key certificates [3]. IBDVS schemes have various cryptographic applications such as licensing software, auctions, and electronic voting.

However, in most practical scenarios, a designated verifier does not artificially generate a (designated verifier) signature for non-transferability using a simulation algorithm. Based on this belief, an adversary capturing a signature first transmitted

---

Manuscript received Sept. 20, 2011; revised Dec. 2, 2011; accepted Dec. 15, 2011.

This work was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology in Korea (Grant No. 2011-0029925, Development of Data Privacy Enhancing Technologies in Future Complex Computing Environments).

JuHee Ki (phone: +82 10 2719 6031, eye@keit.re.kr) is with the Information Security PD Team, KEIT, Daejeon, Rep. of Korea.

Jung Yeon Hwang (videmot@etri.re.kr) and Beom-Hwan Chang (bchang@etri.re.kr) are with the Cyber Security-Convergence Research Department, ETRI, Daejeon, Rep. of Korea.

DaeHun Nyang (nyang@inha.ac.kr) is with the Information Security Research Laboratory, Inha University, Incheon, Rep. of Korea.

Dong Hoon Lee (donghlee@korea.ac.kr) and Jong-in Lim (jilim@korea.ac.kr) are with CIST, Korea University, Seoul, Rep. of Korea.

http://dx.doi.org/10.4218/etrij.12.0111.0597

from a party would imply that the signature originated from the sender, not a receiver, that is, a designated verifier. This helps the adversary to decide who made a signature and to collect critical information, such as the signer's intention. To remedy the privacy problem, a notion of *strongness* for a DVS scheme has been introduced [1], [4]. Under this strongness notion, public verifiability on a DVS is no longer permitted. Instead, only a designated verifier can check the validity of a signature using his or her secret key.

To enhance the privacy of an IBDVS, we allege that even the signer must not be able to verify the validity of its own signatures unless the signer saves signatures in its storage. By this property, even the adversary who has a signer's secret key is not able to verify signatures. We call this privacy property *self-unverifiability* because even the signer itself cannot verify its own signatures, and the property will strictly separate capabilities of signing key and verifying key. This property is quite necessary because it is unfair to a verifier having the exclusive right to verify signatures when a signer mistakenly or intentionally loses its secret signing key to give the right to others who obtain the signing key. In reality, signers frequently lose their signing keys due to computer viruses, malicious software, misconfigurations of related systems, and lost/stolen portable devices. Therefore, without the self-unverifiability, a signer's poor management of its keys might infringe upon the designated verifier's exclusive right of verifiability.

To enhance the privacy of the signer by providing self-unverifiability, we propose a generic method that constructs strong IBDVS schemes with self-unverifiability as well as all the functionalities of IBDVS. We also prove the security of this method. Our design idea is to combine identity-based key encapsulation mechanism (IBKEM) and non-interactive identity-based key sharing (IBKS) schemes. In the generic method, we can flexibly and independently combine any pair of IBKEM and IBKS schemes irrespective of their underlying structures or hardness assumptions. For example, an integer factorization-based IBKEM and a pairing-based IBKS can be combined together. The IBKEM is used to achieve exclusive verifying capability, and the IBKS is used for signing. Accordingly, a designated verifier has two capabilities, decrypting (that is, verifying) and signing, and a signer has only the signing capability. In addition to flexibility, our scheme instantiated from the generic construction can output shorter signatures than that of existing schemes [5], without adding computational overhead.

Various IBDVS schemes have been suggested to achieve strongness [5]-[7]. Several IBDVS schemes rely on a structure to hide a signature using a key that both a signer and a designated verifier (non-interactively) share [6], [8]-[10]. The key can be computed using a signer's or verifier's static long-term secret key. Applying the notion of a keyed hash function to a DVS scheme, one of the schemes proposes a novel method to offer very short signatures [6]. However, these schemes fail to achieve self-unverifiability because they are based upon the static structure of an IBKS method.

One of the promising applications is virus-free software distribution, where a software company will provide validity of signatures for corresponding software but only to clients who buy this service [1]. Pirated software cannot be validated correctly, even when a legal buyer of the software sends it with his/her validation key, because the validation key can be used for generating the signature for the software modified for some malicious purpose.

The remainder of this paper is organized as follows. In section II, we briefly review the strong IBDVS schemes in [6], [11] and show that these schemes have security vulnerability in terms of self-unverifiability. In section III, a formal security model is presented. In section IV, we propose a generic method of constructing an IBDVS scheme with self-unverifiability and prove the security of this method. Finally, in section V, we give concluding remarks.

II. Vulnerabilities in IBDVS Schemes

Using various cryptographic techniques, several IBDVS schemes have been suggested to achieve strongness in verifiability [5]-[7]. Some of them use a specific non-interactive IBKS method between a signer and a designated verifier [6], [8]-[10], where a key is computed with either a signer's or verifier's static long-term secret key and used for a keyed hash function.

We show that any IBDVS scheme that uses the specific structure does not have the self-unverifiability. To illustrate our idea, we briefly review the IBDVS scheme in [6]. The scheme is described as follows:

- **Setup**. Let G and $\mathrm{G_T}$ be additive and multiplicative groups, respectively. Let e: $\mathrm{G}\times\mathrm{G}\rightarrow\mathrm{G_T}$ be a bilinear map, where G and $\mathrm{G_T}$ have prime order q. Let P be a random generator of G, $s(\in Z^*_q)$ is chosen at random, and $P_{\mathrm{pub}}=sP$. The algorithm selects two collision-resistant cryptographic hash functions, $\mathrm{H}_0\colon\{0, 1\}^*\rightarrow\mathrm{G}$ and $\mathrm{H}_1\colon\{0, 1\}^*\rightarrow Z^*_q$. It outputs the master secret key, $msk=s$, and the public scheme parameters, $params=(\mathrm{G}, \mathrm{G_T}, q, e, P, P_{\mathrm{pub}}, \mathrm{H}_0, \mathrm{H}_1)$.
- **KeyExtract**. To extract a decryption key for identity $ID\in\{0, 1\}^*$, return $sk_{ID}=sQ_{ID}$, where $Q_{ID}=\mathrm{H}_0(ID)$.
- **IDSign**. To sign a message $m\in\{0, 1\}^*$ for a designated verifier Bob, Alice computes $Q_{ID_\mathrm{B}}=\mathrm{H}_0(ID_\mathrm{B})\in\mathrm{G}$, $k=e(Q_{ID_\mathrm{B}}, sk_{ID_\mathrm{A}})\in\mathrm{G_T}$, and $\sigma=\mathrm{H}_1(m\,\|\,k)$. The signature on a message m is σ.
- **IDVerify**. To verify the validity of a signature σ on a

message m, the designated verifier Bob computes $Q_{ID_A} = H_0(ID_A)$, $k = e(Q_{ID_A}, sk_{ID_B})$ and tests if $H_1(m \| k) \overset{?}{=} \sigma$ holds. If the equality holds, then it outputs True; otherwise, it outputs False.

To achieve the strong designated verifiability, the above IBDVS scheme takes a simple approach, which is to use a key (non-interactively) shared between a signer and a designated verifier to authenticate a message. This method can be viewed as a keyed hash function, that is, a standard MAC.

Although the scheme yields a short signature, the static structure of the IBKS is vulnerable to exposure of the signer's secret key. This can be easily checked as follows. Assume that an adversary F obtains a signer's secret key, sk_{ID_A}. For a given signature σ on a message m, F can compute $k' = e(Q_{ID_B}, sk_{ID_A}) = e(Q_{ID_B}, Q_{ID_A})^s$ and then check the validity of the signature by $\sigma = H_1(m \| k')$.

A similar weakness exists in the recent IBDVS schemes [9]-[11]. As illustrated above, the weakness is mainly caused by a static structure of IBKS between a signer and a designated verifier.

III. Security Model for an IBDVS Scheme

In this section, we present a formal security model for an IBDVS scheme. In particular, we newly introduce a formal notion of *self-unverifiability.*

1. Identity-Based Designated Verifier Signature Scheme

An IBDVS scheme consists of the following algorithms.

- Setup(1^k). It takes as input a security parameter 1^k, and then outputs the master secret key *msk* and its corresponding public parameters *pp*.

- KeyExtract(*msk*, *ID*). It takes as inputs the master secret key *msk* and an identity *ID*, and then outputs a private signing key sk_{ID}.

- IDSign((sk_S, ID_V), *m*). It takes as inputs a private signing key sk_S, the identity of a designated verifier ID_V, and a message m, and then outputs a signature σ.

- IDVrfy(σ, (ID_S, sk_V, ID_V), *m*). It takes as inputs a signature σ, the identities of a designated verifier and a signer (ID_S, ID_V), a private signing key sk_V, and a message m, and then outputs 1 (Valid) or 0 (Invalid).

2. Security Model

We consider three secuirty properties for an IBDVS scheme: unforgeability, non-transferability, and self-unverifiability. As noted in the literature [12], unforgeability and non-transferability correspond to "unforgeability" and "anonymity" for a ring signature with a ring of two members, respectively.

Unforgeability. Informally, this notion means that any party who cannot access private keys of a signer and a designated verifier is not able to generate a signature. Next, we formally define the notion of unforgeability.

An IBDVS scheme $\sum$ is said to be existentially unforgeable under chosen identity-message attacks (CIMA) if no probabilistic polynomial-time (PPT) adversary F has a non-negligible advantage in the following game: For a security parameter k, a challenger C runs **SetUp** to obtain the master secret key *msk* and its corresponding public parameters *pp*; an adversary F gets the public parameters; and the adversary F is allowed to access to the following **Sign**, **Extract**, and **IDVrfy** oracles to make polynomially-many queries adaptively. Here, "adaptively" means that a query may depend on answers to the previous queries.

- Sign. On a query $<(ID_S, ID_V), m>$, return $\sigma \leftarrow \mathbf{Sign}(sk_S, ID_V, m)$.

- Extract. On a query $<ID>$, return $sk_{ID} \leftarrow \mathbf{KeyExtract}(msk, ID)$.

- IDVrfy. On a query $<\sigma, (ID_S, ID_V), m>$, return $b \leftarrow \mathbf{IDVrfy}(\sigma, (ID_S, sk_V, ID_V), m)$.

Finally, F outputs $((ID_S, ID_V), m', \sigma')$. Assume that σ' on $((ID_S, ID_V), m')$ is valid, that is, $1 \leftarrow \mathbf{IDVrfy}(\sigma', (ID_S, ID_V), m')$.

F succeeds in the above game if the following two conditions hold; i) any of ID_S and ID_V has not been queried to **Extract** oracle and ii) the $((ID_S, ID_V), m')$ tuple is not the same as any of the tuples queried to **Sign** oracle. The event of the success is denoted by $\mathrm{Suc}_{\mathrm{Forg}}$. The EUF-CIMA advantage of F for $\sum$ is defined by $\mathsf{Adv}_{F,\sum}^{\text{EUF-CIMA}}(k) = \Pr[\mathrm{Suc}_{\mathrm{Forg}}]$.

Non-transferability. Informally, non-transferability means that any third party except a signer and a designated verifier cannot identify the real generator of a DVS. An IBDVS scheme $\sum$ is said non-transferable if there exists no PPT adversary that has a non-negligible advantage to distinguish the distribution of signatures generated from real executions of the scheme (with a secret signing key) and that of signatures from the simulator **Sim**. Here, **Sim** takes as input $((ID_S, sk_V), m)$, and then outputs a simulated signature. More specifically, we consider the following game: For a security parameter k, a challenger C runs **SetUp** to obtain the master secret key *msk* and its corresponding public parameters *pp*; an adversary F gets the public parameters; and the adversary F is given access to the following oracles to make polynomially-many queries adaptively. Here, "adaptively" means that a query may depend on answers to the previous queries.

- Sign. On a query $<(ID_S, ID_V), m>$, return $\sigma' \leftarrow \mathbf{Sign}(sk_S, ID_V, m)$.

- Extract. On a query $<ID>$, return $sk_{ID} \leftarrow \mathbf{KeyExtract}(msk, ID)$.

- **IDVrfy**. On a query $<\sigma, (ID_S, ID_V), m>$, return b←**IDVrfy**$(\sigma, (ID_S, sk_V, ID_V), m)$.

When F submits $((ID_S, ID_V), m)$ as a challenge, the challenger C picks a bit $b \in \{0, 1\}$ uniformly at random. If b=0, then return σ←**Sign**(sk_S, ID_V, m); otherwise, return σ←**Sim**$((ID_S, sk_V), m)$. The signature σ is given to the adversary F. Finally, F outputs a guess bit b'.

F succeeds in the above game if $b=b'$. The event of the success is denoted by Suc_{NT}. The advantage of F for $\sum$ is defined by $\mathsf{Adv}_{F,\sum}^{\text{Non-Trans}}(k)=\Pr[Suc_{NT}]$.

Alternatively, we can define this notion using the "anonymity" for a ring signature with a ring of two members [12].

Self-unverifiability. The notion of *signature privacy* means that the validity of a signature associated with a designated verifier should be confirmed only with the designated verifier's secret key, where the adversary does not have an access to the signing key. This notion is also known as *strongness* in the literature [1], [4]. To enhance the signature privacy, we introduce a stronger notion called *self-unverifiability* that allows an adversary to access even a signing key. In other words, self-unverifiability captures that the validity of a signature associated with a designated verifier should be confirmed only with the designated verifier's secret key, even when a signing key to be used to generate the signature is exposed in the future. As shown in section II, some IBDVS schemes achieve only signature privacy, not self-unverifiability. Next, we formally define this notion.

For a security parameter k, a challenger C runs **SetUp** to obtain the master secret key msk and its corresponding public parameters pp. An adversary F gets the public parameters. The adversary F is given access to the following oracles to make polynomially-many queries adaptively. Here, "adaptively" means that a query may depend on answers to the previous queries.

- **Extract**. On a query $<ID>$, return sk_{ID}←**KeyExtract**(msk, ID).

- **Sign**. On a query $<(ID_S, ID_V), m>$, return σ'←**Sign**(sk_S, ID_V, m).

- **IDVrfy**. On a query $<\sigma, (ID_S, ID_V), m>$, return b←**IDVrfy** $(\sigma, (ID_S, sk_V, ID_V), m)$.

When F submits $((ID_S, ID_V), m, m')$ as a challenge, the challenger C picks a bit $b \in \{0, 1\}$ uniformly at random. If b=0, then return σ←**Sign**(sk_S, ID_V, m); otherwise, return σ←**Sign**(sk_S, ID_V, m'). The signature σ is given to the adversary F. We assume that F is already aware of the private signing key sk_S by querying ID_S to **Extract** oracle. Finally, F outputs a guess bit b'.

F succeeds in the above game if (1) $b=b'$, (2) ID_V has never been queried to **Extract** oracle, and (3) neither $(\sigma, (ID_S, ID_V), m)$ nor $(\sigma, (ID_S, ID_V), m')$ have been queried to **IDVrfy**. The event of the success is denoted by Suc_{FSP}. The advantage of F for $\sum$ is defined by $\mathsf{Adv}_{F,\sum}^{FSP}(k)=\Pr[Suc_{FSP}]$.

IV. Our Generic Approach for Self-Unverifiability

To overcome the problem presented in section II, we propose a generic method based on an IBKEM scheme and a non-interactive two-party IBKS scheme. An IBKEM scheme is an identity-based variant of an ordinary KEM [13], [14].

1. Two Cryptographic Primitives

Before describing our method in detail, we first review an IBKEM scheme [14] and a non-interactive two-party IBKS scheme that are used as the building blocks for our construction.

An IBKEM scheme consists of four algorithms: KEM-Setup, KEM-Ext, KEM-Enc, and KEM-Dec.

- KEM-Setup. It takes as input a security parameter 1^k, and then outputs a master secret key msk_{KEM} and its corresponding public parameter pp_{KEM}. K_D is a plaintext space associated with pp_{KEM}.

- KEM-Ext. It takes as inputs the master secret key msk_{KEM} and an identity ID, and then outputs a secret key $sk_{KEM,ID}$.

- KEM-Enc. It is a PPT algorithm that on inputs of pp_{KEM} and an identity ID outputs a random 'one-time' key $k_D \in K_D$ and its ciphertext θ.

- KEM-Dec. It is a deterministic algorithm that on inputs of a private key $sk_{KEM,ID}$ and a ciphertext θ outputs a key k_D.

Basically, it is required that an IBKEM scheme should satisfy the correctness, that is, for given (msk_{KEM}, pp_{KEM}) ←KEM-Setup(1^k), for any identity ID, $sk_{KEM,ID}$←KEM-Ext(msk_{KEM}, ID), and (k_D, θ)←KEM-Enc(pp_{KEM}, ID), we have k_D←KEM-Dec$(sk_{KEM,ID}, \theta)$. We say that an IBKEM scheme IBKEM is semantically-secure if no PPT adversary can gain a non-negligible advantage to guess a bit b for given k_b, θ, where (k_D, θ)←KEM-Enc(pp_{KEM}, ID), b is a randomly selected bit, and if b=0, then $k_b=k_D$; otherwise, if b=1, k_b is a random number. Here, we assume that ID is chosen by an adversary.

Many identity-based encryption schemes can be represented in the IBKEM/DEM framework [14]. As an example for an IBKEM scheme, we can consider the Boneh-Franklin KEM [15], where KEM-Enc is defined by $(k_D=e(Q_{ID}, P_{pub})^r, \theta=rP)$← KEM-Enc$(pp_{KEM}, ID)$ for random $r \in Z^*_q$.

A (non-interactive) two-party IBKS scheme consists of three algorithms: KS-Setup, KS-Ext, and sKS.

- KS-Setup. It takes as input a security parameter 1^k and then outputs a master secret key msk_{KS} and its corresponding public parameter pp_{KS}.

- KS-Ext. It takes as inputs the master secret key msk_{KS} and an identity ID and then outputs a secret key $sk_{\mathrm{KS},ID}$.

- sKS. It is a deterministic algorithm that takes as inputs a user U_{S}'s secret key $sk_{\mathrm{KS},ID_{\mathrm{S}}}$ and a user U_{V}'s public identity ID_{V} and then outputs a key TK. The algorithm has symmetry of computation for the participants. That is, given the user U_{V}'s secret key $sk_{\mathrm{KS},ID_{\mathrm{V}}}$ and the user U_{S}'s public identity ID_{S}, it outputs the same key TK. That is, TK=sKS(ID_{V}, $sk_{\mathrm{KS},ID_{\mathrm{S}}}$) =sKS($ID_{\mathrm{S}}$, $sk_{\mathrm{KS},ID_{\mathrm{V}}}$).

Basically, it is required that an IBKS scheme should satisfy the correctness; that is, for any ID_{V} and ID_{S}, $sk_{\mathrm{KS},ID_{\mathrm{S}}}$←KS-Ext($ID_{\mathrm{S}}$), $sk_{\mathrm{KS},ID_{\mathrm{V}}}$←KS-Ext($ID_{\mathrm{V}}$), we have TK=sKS(ID_{V}, $sk_{\mathrm{KS},ID_{\mathrm{S}}}$)=sKS($ID_{\mathrm{S}}$, $sk_{\mathrm{KS},ID_{\mathrm{V}}}$).

We say that an IBKS scheme IBKS is key-indistinguishable if no PPT adversary can gain a non-negligible advantage in the following experiment with a simulator S: Initially, S sets up system parameters for IBKS and gives the resulting public parameters to the adversary F. Also, a secret key is properly defined for each user. Next, S simulates an attack environment for the IBKS scheme by providing Execute and Reveal queries. The adversary F can get transcripts of an honest execution of IBKS or a common key computed from an execution of IBKS according to queries. When F issues Test query for a pair of two identities, (ID_1, ID_2), S returns a value K_b after selecting a random bit $b\in\{0, 1\}$, where K_0 is a key generated from an honest execution of IBKS with (ID_1, ID_2) and K_1 is a random number selected from a key space.

In contrast to normal IBKS schemes [16], the above IBKS scheme does not require any communication between two participants for sharing a key. As an example for a non-interactive IBKS scheme, we can consider the two-party IBKS scheme in [16], [17], which is essentially the same as that in the previous section in that a shared key in the IBKS is defined by $e(Q_{ID_{\mathrm{V}}}, sk_{\mathrm{KS},ID_{\mathrm{S}}})=e(Q_{ID_{\mathrm{V}}}, Q_{ID_{\mathrm{S}}})^s=e(sk_{\mathrm{KS},ID_{\mathrm{V}}}, Q_{ID_{\mathrm{S}}})$.

2. Construction

We present a generic way to construct an IBDVS scheme with self-unverifiability as follows: Assume that an IBKEM scheme IBKEM=(KEM-Setup, KEM-Ext, KEM-Enc, KEM-Dec) and a non-interactive two-party IBKS scheme IBKS=(KS-Setup, KS-Ext, sKS) are given. Let $\mathrm{H}:\{0, 1\}^* \rightarrow \{0, 1\}^\lambda$ be a cryptographic hash function.

- Setup. It takes as input a security parameter 1^k and then performs the setup algorithms for IBKEM and IBKS, respectively, that is, (msk_{KEM}, pp_{KEM})←KEM-Setup(1^k) and (msk_{KS}, pp_{KS})←KS-Setup(1^k). The master secret key is msk=(msk_{KEM}, msk_{KS}), and its corresponding public parameter is pp=(pp_{KEM}, pp_{KS}).

- KeyExtract. It takes as input an identity ID and then performs the key extract algorithms for IBKEM and IBKS, that is, $sk_{\mathrm{KEM},ID}$←KEM-Ext(msk_{KEM}, ID) and $sk_{\mathrm{KS},ID}$←KS-Ext(msk_{KS}, ID). A secret key for the identity ID is sk_{ID}=($sk_{\mathrm{KEM},ID}$, $sk_{\mathrm{KS},ID}$).

- IDSign. It is a PPT algorithm that takes as inputs a message $m\in\{0, 1\}^*$, verifier's identity ID_{V}, and signer's secret key $sk_{ID_{\mathrm{S}}}$ and then first computes TK←sKS(ID_{V}, $sk_{\mathrm{KS},ID_{\mathrm{S}}}$). It also computes ($k_{\mathrm{D}}$, θ)←KEM-Enc(ID_{V}), η←H(k_{D}, TK), and τ←H($\eta \| \theta \| m$). The signature on a message m is σ←(θ, τ).

- IDVerify. It takes as inputs a signature σ←(θ, τ), message m, and verifier's secret key $sk_{ID_{\mathrm{V}}}$=($sk_{\mathrm{KEM},ID_{\mathrm{V}}}$, $sk_{\mathrm{KS},ID_{\mathrm{V}}}$) and then computes TK'←sKS(ID_{S}, $sk_{\mathrm{KS},ID_{\mathrm{V}}}$), k'_{D}←KEM-Dec(pp_{KEM}, $sk_{\mathrm{KEM},ID_{\mathrm{V}}}$, θ), and η'←H(k'_{D}, TK'). It tests if $\mathrm{H}(\eta' \| \theta \| m) \overset{?}{=} \tau$ holds. If the equality holds, then it outputs Valid; otherwise, it outputs Invalid.

In the above construction, the use of the keys, k_{D} and TK, are intended to provide two security properties. The key k_{D} is used to achieve the self-unverifiability, which means that exposure of a signer's secret key does not compromise a MAC key η, which is used in generation of a signature. Obviously, it will be intractable to compute the key η=H(k_{D}, TK) because it will be intractable to compute k_{D} without knowledge of $sk_{ID_{\mathrm{V}}}$ if the semantic security of a given IBKEM is guaranteed. The key TK is used to achieve unforgeability against outside attackers except a signer and a designated verifier.

In Fig. 1, we show a schematic diagram for our generic construction of an IBDVS from an IBKEM scheme and a non-interactive two-party IBKS scheme.

Remark 1. We note that our generic approach intrinsically provides delegatability [18], which can be used for delegating signing capability. In other words, if a long static key TK shared between a sender and a receiver is given to a delegate, he or she can make a signature on behalf of the signer.

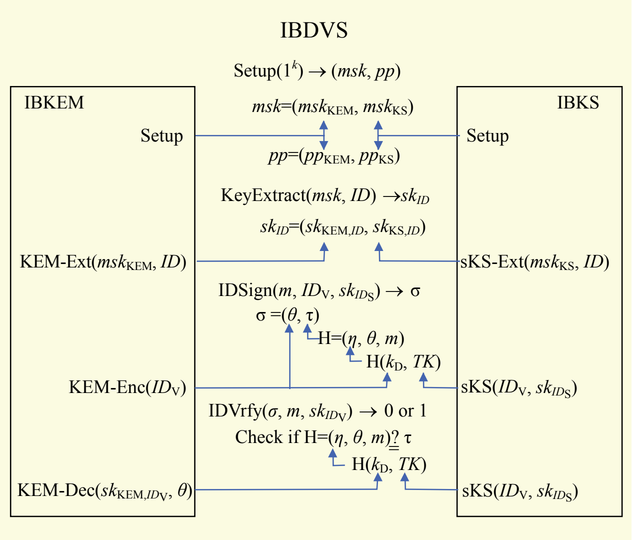


Fig. 1. Generic construction of IBDVS.

3. Security

We prove that the IBDVS scheme constructed above achieves existential-unforgeability, non-transferability, and self-unverifiability.

Theorem 1. If a given IBKS scheme is key-indistinguishable, then the above IBDVS scheme is existentially unforgeable in the random oracle model.

Proof. We show that we can build an efficient distinguisher D attacking the key-indistinguishability of the underlying IBKS scheme by using a PPT forger F attacking the IBDVS scheme constructed from the generic method.

Assume that the distinguisher D is given a public parameter pp_{KS}. Also, assume that q_{E} and q_{S} are the numbers of Extract and Sign queries that an adversary can make to **Extract** and **Sign** oracles, respectively. D first picks α and β uniformly from $\{1,\ldots, q_{\mathrm{E}}+q_{\mathrm{S}}\}$ for two target identities that F will output as a final forgery. Let ID_1^* and ID_2^* denote the α-th and β-th new identities that are queried to either the **Extract** or **Sign** oracle, respectively. D issues Test query for a pair of two identities (ID_1^*, ID_2^*), and then gets back a challenge value K_b, where K_0 is a key generated from an honest execution of the given IBKS scheme with (ID_1, ID_2) and K_1 is a random number selected from a key space. The goal of the distinguisher D correctly guesses the bit b by running F as a sub-algorithm.

The distinguisher D provides F with a simulation environment for the unforgeability game as follows: Let $\mathrm{F}^{\mathrm{G}} \Rightarrow \pi$ denote the event that an adversary F outputs a forgery π in this unforgeability game; assume that an adversary F makes q_{S}, q_{E}, and q_{V} queries to the **Sign**, **Extract**, and **IDVrfy** oracles, respectively; and we also assume that hash queries are never repeated.

The challenger D first runs the **SetUp** algorithm of IBKEM to generate the master secret key msk_{KEM} and its corresponding public parameters pp_{KEM}. Define $pp=(pp_{\mathrm{KEM}}, pp_{\mathrm{KS}})$ and then give pp to the adversary F. The challenger answers F's oracle queries as follows:

- **Hash**. On a query $<M>$, pick h uniformly at random from $\{0, 1\}^{\lambda}$, and return h.

- **Extract**. On a query $<ID>$, proceed as follows:

(a) If $ID=ID_1^*$ or ID_2^*, then abort the simulation.

(b) Else if, return $sk_{ID}=(sk_{\mathrm{KEM},ID}, sk_{\mathrm{KS},ID})$ by running KEM-Ext, that is, $sk_{\mathrm{KEM},ID} \leftarrow$ KEM-Ext(msk_{KEM}, ID) and querying to the $\mathbf{Extract}_{\mathbf{KS}}$ oracle for KS-Ext, that is, $sk_{\mathrm{KS},ID} \leftarrow \mathbf{Extract}_{\mathbf{KS}}(ID)$, where the $\mathbf{Extract}_{\mathbf{KS}}(ID)$ is the oracle to extract a secret key corresponding ID for KS-Ext of the IBKS scheme.

- **Sign**. On a query $<(ID_{\mathrm{S}}, ID_{\mathrm{V}}), m>$, proceed as follows:

(a) If $ID_{\mathrm{S}}=ID_1^*$ and $ID_{\mathrm{V}}=ID_2^*$, then compute $(k_{\mathrm{D}}, \theta) \leftarrow$ KEM-Enc(ID_{V}) and (η, τ) by querying $<k_{\mathrm{D}}, TK>$ and $<\eta, \theta, m>$ to the hash oracle H, that is, $\eta \leftarrow \mathrm{H}(k_{\mathrm{D}}, TK)$ and $\tau \leftarrow \mathrm{H}(\eta, \theta, m)$, where TK is the challenge that is given to D for IBKS in advance. Finally, return $\sigma \leftarrow (\theta, \tau)$.

(b) Else if $ID_{\mathrm{S}} \neq ID_1^*$ and $ID_{\mathrm{V}} \neq ID_2^*$, then compute $TK \leftarrow$ sKS$(ID_{\mathrm{V}}, sk_{\mathrm{KS},ID_{\mathrm{S}}})$ and $(k_{\mathrm{D}}, \theta) \leftarrow$ KEM-Enc(ID_{V}) and also (η, τ) by querying $<k_{\mathrm{D}}, TK>$ and $<\eta, \theta, m>$ to the hash oracle H, that is, $\eta \leftarrow \mathrm{H}(k_{\mathrm{D}}, TK)$ and $\tau \leftarrow \mathrm{H}(\eta, \theta, m)$, respectively. Finally, return $\sigma \leftarrow (\theta, \tau)$.

(c) Else if $(ID_{\mathrm{S}} \neq ID_1^* \wedge ID_{\mathrm{V}}=ID_2^*)$ or $(ID_{\mathrm{S}}=ID_1^* \wedge ID_{\mathrm{V}} \neq ID_2^*)$, then compute TK by computing $TK=$sKS$(ID_{\mathrm{V}}, sk_{\mathrm{KS},ID_{\mathrm{S}}})$ or sKS$(ID_{\mathrm{S}}, sk_{\mathrm{KS},ID_{\mathrm{V}}})$. Note that we have $TK=$sKS$(ID_{\mathrm{S}}, sk_{\mathrm{KS},ID_{\mathrm{V}}})$ $=$sKS$(ID_{\mathrm{V}}, sk_{\mathrm{KS},ID_{\mathrm{S}}})$ by the correctness of the given IBKS scheme. Then, compute $(k_{\mathrm{D}}, \theta) \leftarrow$ KEM-Enc(ID_{V}) and also (η, τ) by querying $<k_{\mathrm{D}}, TK>$ and $<\eta, \theta, m>$ to the hash oracle H, that is, $\eta \leftarrow \mathrm{H}(k_{\mathrm{D}}, TK)$ and $\tau \leftarrow \mathrm{H}(\eta, \theta, m)$, respectively. Finally, return $\sigma \leftarrow (\theta, \tau)$.

- **IDVrfy**. On a query $<\sigma, (ID_{\mathrm{S}}, ID_{\mathrm{V}}), m>$, proceed as follows:

(a) If $ID_{\mathrm{S}}=ID_1^*$ and $ID_{\mathrm{V}}=ID_2^*$, then check the validity of the given query by using the pre-images of $\sigma=(\theta, \tau)$ and return b.

(b) Else if $ID_{\mathrm{S}} \neq ID_1^*$ and $ID_{\mathrm{V}} \neq ID_2^*$, then return the result of the **IDVrfy**, that is, $b \leftarrow \mathbf{IDVrfy}(\sigma, (ID_{\mathrm{S}}, sk_{\mathrm{V}}, ID_{\mathrm{V}}), m)$.

(c) Else if $(ID_{\mathrm{S}} \neq ID_1^* \wedge ID_{\mathrm{V}}=ID_2^*)$ or $(ID_{\mathrm{S}}=ID_1^* \wedge ID_{\mathrm{V}} \neq ID_2^*)$, then according to the third condition in the above signing oracle, check the validity of the given query and then return b.

Finally, F outputs $\pi=((ID_{\mathrm{S}}, ID_{\mathrm{V}}), m', \sigma')$ and then D outputs $b \leftarrow \mathbf{IDVrfy}(\sigma', (ID_{\mathrm{S}}, ID_{\mathrm{V}}), m')$. Let Abort denote the event that an abortion occurs in the above game and $\sim$Abort the negation of Abort. We have $\Pr[\mathrm{F}^{\mathbf{G}} \Rightarrow \pi]=\Pr[\mathrm{F}^{\mathbf{G}} \Rightarrow \pi \wedge (\sim\text{Abort} \vee \text{Abort})]=\Pr[\mathrm{F}^{\mathbf{G}} \Rightarrow \pi \wedge \sim\text{Abort}]+\Pr[\mathrm{F}^{\mathbf{G}} \Rightarrow \pi \wedge \text{Abort}]$ $=\Pr[\mathrm{F}^{\mathbf{G}} \Rightarrow \pi \mid \sim\text{Abort}] \cdot \Pr[\sim\text{Abort}]+\Pr[\mathrm{F}^{\mathbf{G}} \Rightarrow \pi \wedge \text{Abort}]$. The first, second, and third equalities hold by the equivalent expansion.

We have $\Pr[\mathrm{F}^{\mathbf{G}} \Rightarrow \pi \wedge \text{Abort}]=0$ because Abort means that the simulation is aborted and so the forger could not output $ID_{\mathrm{S}}=ID_1^*$ and $ID_{\mathrm{V}}=ID_2^*$ for a final forgery. Note that Abort does not occur if α and β are correctly guessed because the remaining parameters are identically distributed, and there will be no meaningful relation among random hash outputs and signatures except the given challenge key TK. The probability of the correct guess is $\Pr[\sim\text{Abort}] \leq 2/q_{\mathrm{E}}(q_E-1)$. If the given challenge TK is correct, that is, $TK=$sKS$(ID_{\mathrm{V}}, sk_{\mathrm{KS},ID_{\mathrm{S}}})$, then the presented simulation is perfect. This means that forging on the target identities was successful and the resulting forgery was valid at least with the advantage of the forger. However, if TK was a random key, then the forger would get a negligible advantage in forging on the target identities. Let $\varepsilon_{\mathrm{IBKS}}$ be the (maximum) advantage of D attacking the IBKS with respect to key-indistinguishability. Thus, we have $\Pr[\mathrm{F}^{\mathbf{G}} \Rightarrow \pi \mid \sim\text{Abort}]=\varepsilon_{\mathrm{IBKS}}$. So, we have $\Pr[F^{\mathbf{G}} \Rightarrow \pi]=\Pr[\mathrm{F}^{\mathbf{G}} \Rightarrow \pi \mid \sim\text{Abort}] \cdot \Pr[\sim\text{Abort}] \leq$

$\varepsilon_{\mathrm{IBKS}}\cdot 2/q_{\mathrm{E}}(q_{\mathrm{E}}-1)$. Hence, if $\varepsilon_{\mathrm{IBKS}}$ is negligible, then $\Pr[\mathrm{F}^{\mathbf{G}}\Rightarrow\pi]$ is negligible. □

Theorem 2. If a given IBKS scheme is correct, then the above IBDVS scheme is non-transferable.

Proof. We show that there exists a simulator Sim such that no PPT adversary has a non-negligible advantage in distinguishing the distribution of signatures generated from real executions of the constructed IBDVS scheme with a signing key and that from the simulator Sim with a designated verifier's secret key. Define Sim by

$$(\theta, \tau)\leftarrow \mathsf{Sim}(ID_{\mathrm{S}}, sk_{ID_{\mathrm{V}}}, m),$$

where for a message $m\in\{0, 1\}^*$, $TK\leftarrow\mathsf{sKS}(ID_{\mathrm{S}}, sk_{\mathrm{KS},ID_{\mathrm{V}}})$, $(k_{\mathrm{D}}, \theta)\leftarrow\mathsf{KEM\text{-}Enc}(pp_{\mathrm{KEM}}, ID_{\mathrm{V}})$, $\eta\leftarrow\mathrm{H}(k_{\mathrm{D}}, TK)$, and $\tau\leftarrow\mathrm{H}(\eta, \theta, m)$. By the correctness property of the given two-party IBKS scheme, we have that $TK=\mathsf{sKS}(ID_{\mathrm{S}}, sk_{\mathrm{KS},ID_{\mathrm{V}}})$ $=\mathsf{sKS}(ID_{\mathrm{V}}, sk_{\mathrm{KS},ID_{\mathrm{S}}})$. So, the distribution of TK is identical in both Sim and IDSign algorithms. In addition, since the simulator uses the same KEM-Enc as in IDSign, the distribution of (k_{D}, θ) is also identical in the Sim and IDSign algorithms. Therefore, $\eta\leftarrow\mathrm{H}(k_{\mathrm{D}}, TK)$ and $\tau\leftarrow\mathrm{H}(\eta, \theta, m)$ are identically distributed in the Sim and IDSign algorithms. □

Theorem 3. If a given IBKEM scheme is semantically secure, then the above IBDVS scheme achieves self-unverifiability in the random oracle model.

Proof. We show that an IBDVS scheme constructed from the generic method presented above achieves self-unverifiability, that is, any PPT adversary gets a negligible advantage in the game for self-unverifiability, which is defined in section III.2. Using the so-called game-playing technique [19], we prove this theorem by considering a sequence of games. The first game is defined for the original self-unverifiability model, and the second game is defined as a modification of the first game, where a random key k_r instead of k_{D} computed in KEM-Enc is used to generate a signature for testing an adversary. For convenience, we denote by **G0** and **G1** the first and the second game in the random oracle, respectively. To complete the proof, we will show that any PPT adversary gets a negligible advantage in the second game, and **G0** and **G1** are identical except a negligible distribution. Let $\mathrm{F}^{\mathrm{G}}\Rightarrow b_{\mathrm{CG}}$ denote the event that an adversary F outputs a correct bit b in game $\mathrm{G}\in\{\mathbf{G0}, \mathbf{G1}\}$. In the games, we assume that an adversary makes q_S, q_{E}, and q_{V} queries to the **Sign**, **Extract**, and **IDVrfy** oracles, respectively. We also assume that hash queries are never repeated.

The game **G0** is the original unforgeability game, which is defined in section III.2 with our specific IBDVS scheme. Next, we define the game more concretely. A challenger C runs the **SetUp** algorithm, that is, the **Setup** algorithms for IBKEM and IBKS to generate the master secret key $msk=(msk_{\mathrm{KEM}}, msk_{\mathrm{KS}})$ and its corresponding public parameters $pp=(pp_{\mathrm{KEM}}, pp_{\mathrm{KS}})$, and then give pp to an adversary F. The challenger answers F's oracle queries as follows:

- **Hash**. On a query $<M>$, pick h uniformly at random from $\{0, 1\}^{\lambda}$, and return h.

- **Extract**. On a query $<ID>$, return $sk_{ID}\leftarrow$**KeyExtract**(msk, ID), that is, $sk_{ID}=(sk_{\mathrm{KEM},ID}, sk_{\mathrm{KS},ID})$ where $sk_{\mathrm{KEM},ID}\leftarrow\mathsf{KEM\text{-}Ext}(msk_{\mathrm{KEM}}, ID)$ and $sk_{\mathrm{KS},ID}\leftarrow\mathsf{KEM\text{-}Ext}(msk_{\mathrm{KS}}, ID)$.

- **Sign**. On a query $<(ID_{\mathrm{S}}, ID_{\mathrm{V}}), m>$, compute $TK\leftarrow\mathsf{sKS}(ID_{\mathrm{V}}, sk_{\mathrm{KS},ID_{\mathrm{S}}})$. It also computes $(k_{\mathrm{D}}, \theta)\leftarrow\mathsf{KEM\text{-}Enc}(ID_{\mathrm{V}})$ and (η, τ) by querying $<k_{\mathrm{D}}, TK>$ and $<\eta, \theta, m>$ to the hash oracle H, that is, $\eta\leftarrow\mathrm{H}(k_{\mathrm{D}}, TK)$ and $\tau\leftarrow\mathrm{H}(\eta, \theta, m)$. Finally, return $\sigma\leftarrow(\theta, \tau)$.

- **IDVrfy**. On a query $<\sigma, (ID_{\mathrm{S}}, ID_{\mathrm{V}}), m>$, return $b\leftarrow$**IDVrfy** $(\sigma, (ID_{\mathrm{S}}, sk_{\mathrm{V}}, ID_{\mathrm{V}}), m)$.

When F submits $((ID_{\mathrm{S}}, ID_{\mathrm{V}}), m, m')$ as a challenge, the challenger C picks a bit $b\in\{0, 1\}$ uniformly at random. If $b=0$, then return $\sigma\leftarrow$**Sign**$(sk_{\mathrm{S}}, ID_{\mathrm{V}}, m)$. Else if $b=1$, choose a random message m' and then return $\sigma\leftarrow$**Sign**$(sk_{\mathrm{S}}, ID_{\mathrm{V}}, m')$. The signature σ is given to the adversary F. We assume that F is already aware of the private signing key sk_{S} by querying ID_{S} to **Extract** oracle. Finally, F outputs a guess bit b'.

The game **G1** is modified from the original game **G0** regarding the **Sign** and **Extract** oracle queries. The modification is described as follows:

- **Extract**. On a query $<ID>$, return $sk_{ID}\leftarrow$**KeyExtract**(msk, ID).

- **Sign**. On a query $<(ID_{\mathrm{S}}, ID_{\mathrm{V}}), m>$, compute $TK\leftarrow\mathsf{sKS}(ID_{\mathrm{V}}, sk_{\mathrm{KS},ID_{\mathrm{S}}})$. It also computes $(k_{\mathrm{D}}, \theta)\leftarrow\mathsf{KEM\text{-}Enc}(ID_{\mathrm{V}})$ and (η, τ) by querying $<k_{\mathrm{D}}, TK>$ and $<\eta, \theta, m>$ to the hash oracle H, that is, $\eta\leftarrow\mathrm{H}(k_{\mathrm{D}}, TK)$ and $\tau\leftarrow\mathrm{H}(\eta, \theta, m)$. Finally, return $\sigma\leftarrow(\theta, \tau)$.

When F submits $((ID_{\mathrm{S}}, ID_{\mathrm{V}}), m, m')$ as a challenge, a bit b is chosen uniformly at random from $\{0, 1\}$. If $b=0$, then proceed as follows: Compute $TK\leftarrow\mathsf{sKS}(ID_{\mathrm{V}}, sk_{\mathrm{KS},ID_{\mathrm{S}}})$ and (k_{D}, θ) $\leftarrow\mathsf{KEM\text{-}Enc}(ID_{\mathrm{V}})$. Pick a random key k_{R}, let $k_{\mathrm{D}}=k_{\mathrm{R}}$, and compute (η, τ) by querying $<k_{\mathrm{D}}, TK>$ and $<\eta, \theta, m>$ to the hash oracle H, that is, $\eta\leftarrow\mathrm{H}(k_{\mathrm{D}}, TK)$ and $\tau\leftarrow\mathrm{H}(\eta, \theta, m)$. Then, return $\sigma=(\theta, \tau)$. Else if $b=1$, choose a random message m' and then return $\sigma\leftarrow$**Sign**$(sk_{\mathrm{S}}, ID_{\mathrm{V}}, m')$. The signature σ is given to the adversary F. We assume that F is already aware of the private signing key sk_{S} by querying ID_{S} to **Extract** oracle. Finally, F outputs a guess bit b'.

We show that an adversary has a negligible advantage in distinguishing **G0** and **G1**. Game **G0** defines k_{D} as the first component of $\mathsf{sKS}(ID_{\mathrm{V}}, sk_{\mathrm{KS},ID_{\mathrm{S}}})$, while game **G1** defines it as a random key. Note that the other parameters of the two games are identically distributed. By assumption, the underlying IBKEM is semantically-secure. Let $\varepsilon_{\mathrm{IBKEM}}$ be the (maximum) advantage of an adversary attacking the IBKEM with respect to the semantic-security. We have $\Pr[\mathrm{F}^{\mathbf{G0}}\Rightarrow b_{\mathrm{CG}}]-\Pr[\mathrm{F}^{\mathbf{G1}}\Rightarrow b_{\mathrm{CG}}]$

$\leq \varepsilon_{\mathrm{IBKEM}}$.

Next, we show that an adversary F succeeds, that is, guesses correctly the challenge bit b in the game **G1** with negligible probability. A signature is defined as a hash output in the game **G1**. In the presented simulation of the random hash function H, it is easy to see that the hash outputs are distributed uniformly at random. Furthermore, the random key k_D is completely unknown from the viewpoint of the adversary by construction. There is no meaningful relation among hash outputs and so signatures. So, the adversary is able to guess the random key with probability $1/\gamma$, where γ is the size of the key space associated with the IBKEM. Typically, for security, γ should be sufficiently large, and so $1/\gamma$ is negligible.

Therefore, summing up the above results, we have $\Pr[F^{G0} \Rightarrow b_{CG}] = (\Pr[F^{G0} \Rightarrow b_{CG}] - \Pr[F^{G1} \Rightarrow b_{CG}]) + \Pr[F^{G1} \Rightarrow b_{CG}] \leq \varepsilon_{\mathrm{IBKEM}} + 1/\gamma$, that is, an adversary has a negligible advantage in the self-unverifiability game. □

4. Instance

To illustrate that our method is effective, we present an example that uses the Boneh-Franklin IBKEM scheme with symmetric bilinear maps [15] and the non-interactive IBKS scheme [17]. The example is described as follows:

- **Setup**. Let G be an additive group and G_T a multiplicative group. Let e: $G \times G \rightarrow G_T$ be a symmetric bilinear map, where G and G_T have prime order q. P is a random generator of G. The algorithm selects $s \in Z^*_q$ at random and computes $P_{pub} \leftarrow sP \in G$. It also selects two collision-resistant cryptographic hash functions, H_0: $\{0, 1\}^* \rightarrow G$ and H: $\{0, 1\}^* \rightarrow \{0, 1\}^{\lambda}$. The algorithm outputs the master secret key, $msk=s$, and its corresponding public parameters, $params=(G, G_T, q, e, P, P_{pub}, H_0, H)$.

- **KeyExtract**. For given identity ID, it computes $Q_{ID} = H_0(ID) \in G$ and $sk_{ID} = sk_{KEM,ID} = sk_{KS,ID} \leftarrow sQ_{ID}$.

- **IDSign**. For given a message $m \in \{0, 1\}^*$, verifier's identity ID_V, and signer's secret key $sk_{ID_S} = sH_0(ID_S)$, it computes $Q_{ID_V} \leftarrow H_0(ID_V) \in G$ and $TK \leftarrow e(sk_{ID_S}, Q_{ID_V}) \in G_T$. It selects $r \in Z^*_q$ and computes $\theta \leftarrow rP \in G$ and $k_d \leftarrow e(rP_{pub}, Q_{ID_V}) \in G_T$. It computes $\eta \leftarrow H(k_d \| TK)$ and $\tau \leftarrow H(\eta \| \theta \| m)$. The signature on a message m is $\sigma=(\theta, \tau)$.

- **IDVerify**. For a given signature $\sigma=(\theta, \tau)$, message m, and verifier's secret key sk_{ID_V}, it computes $Q_{ID_S} \leftarrow H_0(ID_S)$, $TK' \leftarrow e(Q_{ID_S}, sk_{ID_V})$, $k'_D \leftarrow e(\theta, sk_{ID_V})$, and $\eta' \leftarrow H(k'_D \| TK')$. It tests if $H(\eta' \| \theta \| m) \overset{?}{=} \tau$ holds. If the equality holds, then it outputs Valid; otherwise, it outputs Invalid.

Figure 2 shows a concrete instance from our generic construction method using Boneh-Franklin's IBKEM [15] and the non-interactive two-party IBKS scheme of [17].

As shown in [17], the non-interactive IBKS is semantically-secure under the hardness of the decisional bilinear Diffie-Hellman (DBDH) problem, which is one to distinguish whether $t=abc$ or not, for given $(P, aP, bP, cP, e(P, P)^t)$, where e: $G \times G \rightarrow G_T$ is a bilinear map, P is a random generator of G, and $a, b, c \in Z^*_q$. So, the above scheme achieves unforgeability by theorem 1. In addition, as shown in [15], [20], Boneh-Franklin's IBKEM is semantically-secure under the hardness of the computational bilinear Diffie-Hellman (CBDH) problem, which is one to compute $t=abc$ or not, for given (P, aP, bP, cP), and so the DBDH problem. This implies our instance satisfies self-unverifiability by theorem 3.

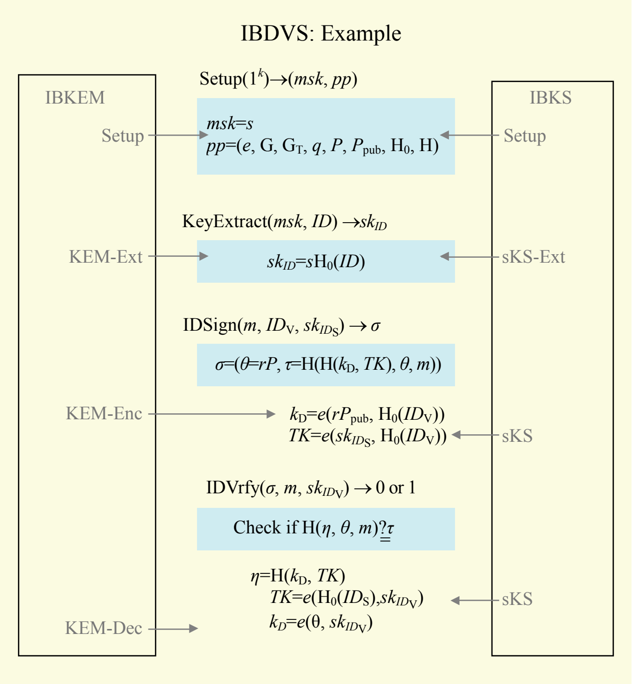


Fig. 2. Example from our generic method.

Table 1. Performance comparison.

	Self-unverifiability	Signature size (bits)	Sign	Verify
[6]	X	160	1P	1P
[5]	O	1,024	$2P + 2E_G$	$1P+1E_G$
[22]	O	1,536	$1P + 3E_G$	$1P+1E_G$
Our generic scheme	O	672	$2P + 2E_G$	2P

Remark 2. A simple combination between two-party identity-based authenticated key agreement schemes [16] and a keyed hash function would not achieve the self-unverifiability of an IBDVS scheme. There is a difference between the security models of our IBDVS scheme and the simple combination scheme. An adversary breaking the self-unverifiability is not allowed to access a designated verifier's

secret key in the security model of our IBDVS scheme but can access the secret key in that of the simple combination scheme. In addition, in contrast to key agreement schemes that typically permit interactive communication between participants, a standard IBDVS scheme should be constructed by one-way transmission from a signer.

5. Comparative Analysis

In this subsection, we compare our scheme in section IV.4 with recent IBDVS schemes in terms of signature length and amount of computation. In this analysis, the hash function H is assumed to output a 160-bit value, that is, λ=160. Our strong IBDVS scheme yields a short signature of which length is almost half of that of the recent strong IBDVS scheme [5]. The DVS scheme of [5] uses a symmetric bilinear pairing map as our scheme. Thus, when 80-bit security level is considered, the 512-bit representation for an element of G and the 1,024-bit representation for an element of G_T should be required for the symmetric bilinear map [14], [16], [21]. Since a signature of [5] consists of an element in G_T, the bit-length of [5] is 1,024 bits while the bit-length of our signature is 672 bits because our signature consists of an element in G and a hash output. [6] does not support self-unverifiability and so is vulnerable to the signer key compromise.

In Table 1, P and E_G represents a pairing computation and a scalar multiplication of the group G, respectively. As shown in Table 1, the computation overhead of our scheme is the same as that of [5]. Finally, when more efficient IBKEM and IBKS are developed, we can simply replace them to obtain more efficient IBDVS with self-unverifiability.

V. Conclusion

We first showed that several recent strong IBDVS schemes do not achieve the self-unverifiability, that is, exposure of a signer's secret keys infringes the designated verifier's exclusive right of verifiability. To overcome the problem, we proposed a generic method to construct IBDVS schemes with the self-unverifiability by combining an identity-based key encapsulation mechanism and a non-interactive identity-based key sharing scheme. Our method instantiates an IBDVS scheme that provides self-unverifiability with short signatures.

In the future, we intend to investigate the construction of an efficient IBDVS with non-delegatability [23] or with more rigorous security notions [24].

References

[1] M. Jakobsson, K. Sako, and R. Impagliazzo, "Designated Verifier Proofs and Their Applications," *Proc. Eurocrypt*, LNCS 1070, 1996, pp.142-154.

[2] F. Laguillaumie and D. Vergnaud, "Designated Verifiers Signature: Anonymity and Efficient Construction from Any Bilinear Map," *Proc. SCN*, LNCS 3352, 2004, pp.107-121.

[3] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes," *Proc. Crypto*, LNCS 196, 1984, pp.47-53.

[4] S. Saeednia, S. Kramer, and O. Markovitch, "An Efficient Strong Designated Verifier Signature Scheme," *Proc. ICISC*, LNCS 2869, 2003, pp. 40-54.

[5] J. Lee, J.K. Chang, and D.H. Lee, "Forgery Attacks on Kang et al.'s Identity-Based Strong Designated Verifier Signature Scheme and Its Improvement with Security Proof," *Comput. Electrical Eng.*, vol. 35, 2009, pp. 49-53.

[6] X. Huang et al., "Short Designated Verifier Signature Scheme and Its Identity-Based Variant," *Int. J. Network Security*, vol. 6, no. 1, 2008, pp. 82-93.

[7] P.K. Kancharla, S. Gummadidala, and A. Sxaena, "Identity-Based Strong Designated Verifier Signature Scheme," *Informatica*, vol. 18, no. 2, 2007, pp. 239-252.

[8] X. Huang et al., "Short (Identity-Based) Strong Designated Verifier Signature Schemes," *Proc. ISPEC*, LNCS 3903, 2006, pp. 214-225.

[9] W. Susilo, F. Zhang, and Y. Mu, "Identity-Based Strong Designated Verifier Signature Schemes," *Proc. ACISP*, LNCS 3108, 2004, pp. 313-324.

[10] J. Zhang and J. Mao, "A Novel ID-Based Designated Verifier Signature Scheme," *Info. Sci.*, vol. 178, 2008, pp. 733-766.

[11] B. Kang, C. Boyd, and E. Dawson, "A Novel Identity-Based Strong Designated Verifier Signature Scheme," *J. Syst. Software*, vol. 178, 2008, pp. 733-766.

[12] A. Bender, J. Katz, and R. Morselli, "Ring Signatures: Stronger Definitions, and Constructions without Random Oracles," *Proc. TCC*, 2007, pp.60-79.

[13] M. Abe, R. Gennaro, and K. Kurosawa, "Tag-KEM/DEM: A New Framework for Hybrid Encryption," *J. Cryptology*, vol. 21, no. 1, 2008, pp. 97-130.

[14] X. Boyen, "A Tapestry of Identity-Based Encryption: Practical Frameworks Compared," *Int. J. Applied Cryptography*, vol. 1, no. 1, 2008, pp. 3-21.

[15] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing," *SIAM J. Comput.*, vol. 32, no. 3, 2003, pp. 586-615.

[16] L. Chen, Z. Cheng, and N.P. Smart, "Identity-Based Key Agreement Protocols from Pairings," *Int. J. Information Security*, vol. 6, no. 4, 2007, pp. 213-241.

[17] R. Sakai, K. Ohgishi, and M. Kasahara, "Cryptosystems Based on Pairing," *Symp. Cryptography Info. Security*, Japan, 2000.

[18] Q. Huang et al., "Efficient Strong Designated Verifier Signature Schemes without Random Oracles or Delegatability," *Cryptology*

ePrint Archive: Report 2009/518, 2009.

[19] M. Bellare and P. Rogaway, "Code-Based Game-Playing Proofs and the Security of Triple Encryption," *Proc. Eurocrypt*, LNCS 4004, 2006, pp. 409-426.

[20] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing," *Proc. Crypto*, LNCS 2139, pp. 213-229, Springer-Verlag, 2001.

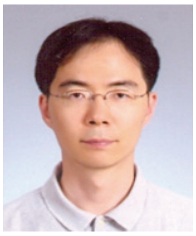
[21] S. Galbraith, K. Paterson, and N. Smart, "Pairings for Cryptographers," *Discrete Applied Mathematics*, vol. 156, no. 16, 2008, pp. 3113-3121.

[22] S. Sun et al., "A New Efficient ID-Based Strong Designated Verifier Signature Scheme," *Proc. 3rd Int. Symp. Info. Sci. Eng.*, 2010, pp. 137-141.

[23] Q. Huang et al., "Efficient Strong Designated Verifier Signature Schemes without Random Oracle or with Non-delegatability," *Int. J. Info. Security,* vol. 10, no. 6, 2011, pp. 373-385.

[24] K. Yoneyama, M. Ushida, and K. Ohta, "Rigorous Security Requirements for Designated Verifier Signatures," *Inscrypt*, LNCS 6584, 2011, pp. 318-335.

JuHee Ki received the BS in mathematics from the University of Seoul and the MS in information security from Korea University, Rep. of Korea, in 2001 and 2003, respectively. Since 2003, she has been a researcher of Korea Evaluation Institute of Industrial Technology (KEIT), Rep. of Korea. Her main research interests include cryptography, broadcast encryption, digital signatures, privacy-enhancing cryptography, identity-based cryptography, attribute-based cryptography, and their applications.



Jung Yeon Hwang received the BS in mathematics from Korea University, the MS and PhD in information security from Korea University, Seoul, in 1999, 2003, and 2006, respectively. He was a post-doctoral researcher at Korea University from 2006 to 2009. Since 2009, he has been a senior member of engineering staff of ETRI, Rep. of Korea. Dr. Hwang's research interests include cryptography, broadcast encryption, digital signatures, identity-based cryptography, privacy-enhancing cryptography, attribute-based cryptography, and their applications.

DaeHun Nyang received the BEng in electronic engineering from Korea Advanced Institute of Science and Technology in 1994, and the MS and PhD in computer science from Yonsei University, Rep. of Korea, in 1996 and 2000, respectively. From 2000 to 2003, he was a senior member of the engineering staff at ETRI, Rep. of Korea. Since 2003, he has been an associate professor in the Computer Information Engineering Department of Inha University, Rep. of Korea, where he is also the founding director of the Information Security Research Laboratory. He is a member of the board of directors and editorial board of Korean Institute of Information Security and Cryptology. Dr. Nyang's research interests include cryptography and information security, privacy, usable security, biometrics and their applications to authentication, and public key cryptography.

Beom-Hwan Chang received his BS, MS, and PhD from Sungkyunkwan University, Rep. of Korea, in 1997, 1999, and 2003, respectively. Since 2003, he has been a senior researcher with the software research laboratory, ETRI, Daejeon, Rep. of Korea. His research interests include network security, programmable network, situation awareness, and security visualization.

Dong Hoon Lee received the BS from the Department of Economics at Korea University, Seoul, in 1985, and the MS and PhD in computer science from the University of Oklahoma, Norman, in 1988 and 1992, respectively. Currently, he is a professor and the vice director of the Graduate School of Information Security at Korea University. His main research interests include the design and analysis of cryptographic protocols in key agreement, encryption, signature, embedded device security, and privacy-enhancing technology.

Jong-in Lim received the BS, MS, and PhD in the Department of Mathematics, Korea University, Seoul, in 1980, 1982, and 1986. Currently, he is a professor and dean of the Graduate School of Information Security, Korea University. Also, he is a former President of the Korea Institute of Information Security and Cryptography. His main research interests include information security policy, cyber warfare, convergence security, privacy, and cryptography.