Anonymous attribute-based designated verifier signature

Olivier Blazy, Laura Brouilhet, Emmanuel Conchon, Mathieu Klingler

Noname manuscript No.
(will be inserted by the editor)

Anonymous Attribute-based Designated Verifier Signature

Olivier Blazy · Laura Brouilhet · Emmanuel Conchon · Mathieu Klingler

Received: date / Accepted: date

Abstract An Attribute-based signature (ABS), is a cryptographic scheme where someone can sign a message using any kind of predicates verified by the attributes he owns. For such scheme, it is expected to be impossible for users to collude to sign a message if none of them is originally able to sign the message on his own. The main advantage of such a solution is that the signer can remain anonymous in the set of users fulfilling the chosen predicate. It can then be used for anonymous authentication for instance.

In this paper, our main contribution is a new designated verifier attribute based signature scheme. In other words, the signer is using his attributes to authenticate a message according to a predicate, and while doing so he can pick another policy such that only users owning attributes fulfilling this policy can check the validity of the signature. It can be used to extend anonymous authentication, ensuring that the designated verifier cannot prove to anyone that a valid authentication has been performed. In addition to classical anonymity, this also increases the privacy of users as no further statistics on valid connection can be deduced.

To do so, we first propose a generic construction of this primitive using standard cryptographic building blocks. An instantiation of this primitive is then described and proved through security games under the Symmetric External Diffie-Hellman (SXDH) assumption. This main contribution is compared to state-of-the-art solutions in terms of both security and efficiency.

Keywords Anonymous Authentification · Attribute-based Cryptography · Signature · Credentials

Olivier Blazy
LIX, CNRS, Inria, École Polytechnique, Institut Polytechnique de Paris, 91120 Palaiseau, France
E-mail: olivier.blazy@polytechnique.edu
ORCID: 0000-0001-6205-8249

Laura Brouilhet
University of Limoges, CNRS, XLIM, UMR 7252, F-87000 Limoges, France
E-mail: laura.brouilhet@unilim.fr

Emmanuel Conchon
University of Limoges, CNRS, XLIM, UMR 7252, F-87000 Limoges, France
E-mail: emmanuel.conchon@unilim.fr
ORCID:0000-0002-6874-5936

Mathieu Klingler
University of Limoges, CNRS, XLIM, UMR 7252, F-87000 Limoges, France
E-mail: mathieu.klingler@unilim.fr
ORCID:0000-0001-7113-2607

1 Introduction

In 1989, Chaum and van Antwerpen introduced the notion of undeniable signature Chaum and Van Antwerpen (1989) that allows signers to control access to their signatures. The signer participates in the verification process to avoid undesirable verifiers knowing about the validity of his signature. However, it is not always efficient because verifiers can collude so that an unauthorized one can obtain information from this interaction (including the validity of the signature). To solve this, Jakobsson *et al.* Jakobsson et al. (1996) introduced designated verifier signature to guarantee that only a specified verifier can verify the signature and can be convinced of its validity. They also introduced a strong designated verifier signature scheme, that implies that the verifier has to use his secret key in the verification phase to avoid signature validation from a third party. The designated verifier can not convince a third party of the validity of the signature and that is due

to the fact that he (the designated verifier) can forge valid signatures. Later this concept was extended by Steinfeld et al. (2003), where they introduced the idea of Universal Designated Verifier Signature, that allows someone given a signature for which he possesses no secret to convert it into a designated verifier signature. Since 2004, several schemes have focused on revisiting these properties like Laguillaumie and Vergnaud (2004); Laguillaumie et al. (2006); Huang et al. (2008); Blazy et al. (2017). However, they do so by relying either on a q-type assumption, on the random oracle, or costly elements in the target group

In 2011, Maji *et al.* introduced Attribute-Based Signature (ABS) Maji et al. (2011). In this kind of signatures, users sign message with their attributes (and an associated predicate) instead of using their secret key. Thereby, the signer is not explicitly identified and so it not possible to link the message to a specific user.

A summary of security properties of each kind of signature is depicted in Table 1.

Using those schemes in practical scenarios requires heavy public key management to designate and verify signatures. To solve this, some works were done with identity-based designated verifier signatures like Susilo et al. (2004) that allows the usage of such protocol by selecting designated verifiers with their identities without managing the keys of those users. For some applications, designating a group of users instead of a single user cannot be achieved easily with identity-based protocol. Attribute-based protocol (ABE) is more suitable for designating groups of users. To achieve this, Fan *et al.* proposed in Fan et al. (2012) a strong attribute-based designated verifier signature. Since then, no other work around Attribute-based Designated Verifier Signature (ABDVS) were done. Our goal is to add a new construction for this primitive that is more efficient and usable in real scenarios.

A classical field of application for attribute-based protocols are cloud-based protocols where the identity of parties involved is less important than their attributes. For instance, the access to health data is often based on a specific attribute or on a specific access policy based on the possession of several attributes. Moreover, with the new GDPR, the privacy of requesters are now more important than ever. With an ABDVS it is therefore possible to propose an anonymous authentication on a cloud server. Indeed, upon reception of the signature, the server is able to authenticate the corresponding entities but it cannot prove this information to a third party. Furthermore thanks to the use of attributes, the identity of the emitter of the signature remains unknown to the cloud ensuring his anonymity. In addition, if the signature is stored by the cloud server, it can be used by researcher to have confidence on the information they are investigating on without being able to disclose or prove anything on these information to an external party.

A direct application of our ABDVS would be for protocols where a proof of valid authentication is damaging. An example would be a broadcast protocol for addresses of Tor bridges that are supposed to be available only to users with a good reputation. It should be impossible for an external authority to coerce a user into proving he is a valid recipient of those addresses, or to coerce them into proving that a broadcast address was validly signed or more generally to convince other users that fake addresses are valid. All those scenarios are encompassed by our security properties.

In this article, we present as a main contribution a new attribute-based designated-verifier signature (ABDVS) which used a transformation from Downgradable IBE to ABE (section 3.2). This transformation allows to switch from a set of attributes to an identity. From his set of attributes, the verifier upgrades his user secret key usk in a new verification key uusk. The new algorithm USKUp executes this modification. uusk is a randomization of a part of usk.

Concerning the security aspect, our construction is unforgeable, non-transferable and respect perfect privacy under the SXDH assumption. We also propose an implementation of our proposed construction and compare it to the Fan et al. (2012) scheme.

2 Definitions

Let ggen be a probabilistic polynomial time (ppt) algorithm that on input $\mathfrak{K}$ returns a description $\mathcal{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$ of asymmetric pairing groups where $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ are cyclic groups of order p for a $\mathfrak{K}$-bit prime p, g_1 and g_2 are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is an efficiently computable (non-degenerated) bilinear map. Define $g_T := e(g_1, g_2)$, which is a generator in $\mathbb{G}_T$.

2.1 Matricial Notation and Assumptions

If $\mathbf{A} \in \mathbb{Z}_p^{(k+1)\times n}$ is a matrix, then $\overline{\mathbf{A}} \in \mathbb{Z}_p^{k\times n}$ denotes the upper matrix of $\mathbf{A}$ and $\underline{\mathbf{A}} \in \mathbb{Z}_p^{1\times n}$ denotes the last row of $\mathbf{A}$.

We use implicit representation of group elements as introduced in Escala et al. (2013). For $s \in \{1, 2, T\}$ and $a \in \mathbb{Z}_p$ define $[a]_s = g_s^a \in \mathbb{G}_s$ as the *implicit representation* of a in $\mathbb{G}_s$ (we use $[a] = g^a \in \mathbb{G}$ if we consider a unique group). More generally, for a matrix

	Authentication	Non-Transferability	Policy-Driven
Digital Signature	✓		
ABS	✓		✓
DVS	✓	✓	
ABDVS	✓	✓	✓

Table 1 Properties Comparison of the various flavors of signature.

$\mathbf{A} = (a_{ij}) \in \mathbb{Z}_p^{n\times m}$ we define $[\mathbf{A}]_s$ as the implicit representation of $\mathbf{A}$ in $\mathbb{G}_s$:

$$[\mathbf{A}]_s := \begin{pmatrix} g_s^{a_{11}} & \dots & g_s^{a_{1m}} \\ & & \\ g_s^{a_{n1}} & \dots & g_s^{a_{nm}} \end{pmatrix} \in \mathbb{G}_s^{n\times m}$$

We will always use this implicit notation of elements in $\mathbb{G}_s$, i.e., we let $[a]_s \in \mathbb{G}_s$ be an element in $\mathbb{G}_s$. Note that from $[a]_s \in \mathbb{G}_s$ it is generally hard to compute the value a (discrete logarithm problem in $\mathbb{G}_s$). Further, from $[b]_T \in \mathbb{G}_T$ it is hard to compute the value $[b]_1 \in \mathbb{G}_1$ and $[b]_2 \in \mathbb{G}_2$ (pairing inversion problem). Obviously, given $[a]_s \in \mathbb{G}_s$ and a scalar $x \in \mathbb{Z}_p$, one can efficiently compute $[ax]_s \in \mathbb{G}_s$. Further, given $[a]_1, [b]_2$ one can efficiently compute $[ab]_T$ using the pairing e. For $\mathbf{a}, \mathbf{b} \in \mathbb{Z}_p^k$ define $e([\mathbf{a}]_1, [\mathbf{b}]_2) := [\mathbf{a}^\top \mathbf{b}]_T \in \mathbb{G}_T$.

We recall the definition of the matrix Diffie-Hellman (MDDH) assumption Escala et al. (2013).

Definition 1 (Matrix Distribution) Let $k \in \mathbb{N}$. We call $\mathcal{D}_k$ a matrix distribution if it outputs matrices in $\mathbb{Z}_p^{(k+1)\times k}$ of full rank k in polynomial time.

Without loss of generality, we assume the first k rows of $\mathbf{A} \xleftarrow{\$} \mathcal{D}_k$ form an invertible matrix. The $\mathcal{D}_k$-Matrix Diffie-Hellman problem is to distinguish the two distributions $([\mathbf{A}], [\mathbf{Aw}])$ and $([\mathbf{A}], [\mathbf{u}])$ where $\mathbf{A} \xleftarrow{\$} \mathcal{D}_k$, $\mathbf{w} \xleftarrow{\$} \mathbb{Z}_p^k$ and $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_p^{k+1}$.

Definition 2 ($\mathcal{D}_k$-Matrix Diffie-Hellman Assumption $\mathcal{D}_k$-MDDH) Let $\mathcal{D}_k$ be a matrix distribution and $s \in \{1, 2, T\}$. We say that the $\mathcal{D}_k$-Matrix Diffie-Hellman ($\mathcal{D}_k$-MDDH) Assumption holds relative to ggen in group $\mathbb{G}_s$ if for all PPT adversaries $\mathcal{D}$,

$$\begin{aligned}\mathbf{Adv}_{\mathcal{D}_k,\mathsf{ggen}}(\mathcal{D}) &\stackrel{\text{def}}{=} |\Pr[\mathcal{D}(\mathcal{G}, [\mathbf{A}]_s, [\mathbf{Aw}]_s) = 1] \\ &\quad - \Pr[\mathcal{D}(\mathcal{G}, [\mathbf{A}]_s, [\mathbf{u}]_s) = 1]| \\ &= \mathsf{negl}(\lambda),\end{aligned}$$

where the probability is taken over $\mathcal{G} \xleftarrow{\$} \mathsf{ggen}(\lambda)$, $\mathbf{A} \xleftarrow{\$} \mathcal{D}_k, \mathbf{w} \xleftarrow{\$} \mathbb{Z}_p^k, \mathbf{u} \xleftarrow{\$} \mathbb{Z}_p^{k+1}$.

In our construction we use the previous definition in the case $k = 1$. In this specific case, the assumption is called Symmetric External Diffie-Hellman assumption (SXDH).

Definition 3 (Symmetric External Diffie-Hellman (SXDH Ateniese et al. (2005))) The SXDH assumption holds in $\mathbb{G}_1$ and $\mathbb{G}_2$ if DDH is hard to compute in both $\mathbb{G}_1$ and $\mathbb{G}_2$.

2.2 Signature Primitives

In this part of the article, we present different signature schemes primitive that we use in our protocol in section 4.2. We recall definition of digital signature and security requirements in the appendix B.

Designated Verifier Signature

Designated verifier signature (DVS) allows a signer to sign a message that only one chosen user can verify. Extended to a set of multiple users is possible. The scheme presented below was introduced by Steinfeld *et.al* in Steinfeld et al. (2003).

Definition 4 Universal designated verifier signature scheme is composed by seven algorithms:

- $\mathsf{Setup}(\mathfrak{K})$: generates the global parameters param of the scheme.
- $\mathsf{KGS}(\mathsf{param})$: outputs a signing key pair for the signer $(\mathsf{sk}_1, \mathsf{pk}_1)$.
- $\mathsf{KGV}(\mathsf{param})$: outputs a verifier key pair for the verifier $\mathsf{sk}_2, \mathsf{pk}_2$.
- $\mathsf{Sign}(\mathsf{sk}_1, m)$ outputs a signature σ verifiable with pk_1.
- $\mathsf{Verify}(\mathsf{pk}_1, m, \sigma)$: checks the validity of σ.
- $\mathsf{Des}(\mathsf{pk}_1, \mathsf{pk}_2, m, \sigma)$: outputs a designated verifier signature $\hat{\sigma}$.
- $\mathsf{DesV}(\mathsf{sk}_2, \mathsf{pk}_1, m, \hat{\sigma})$ checks the validity of the designated signature.

Security requirements for DVS are: unforgeability, DV-unforgeability and non-transferability. This notions are defined in appendix B.

Attribute-Based Signature

The Attribute-Based Signature (ABS) was introduced by Maji et al. (2011). In ABS, a signer with a set of attributes, noted $\mathbb{A}$, can sign a message with a predicate, noted Υ, that is satisfied by his attributes. We note by $\mathcal{U}$ the universe of attributes.

Definition 5 An attribute-based signature scheme consists of four algorithms:

- $\mathsf{Setup}(\mathfrak{K})$: generates public/secret key pair $(\mathsf{mpk}, \mathsf{msk})$.
- $\mathsf{KeyGen}(\mathsf{msk}, \mathbb{A})$: outputs the signing key: $\mathsf{usk}_{\mathbb{A}}$.
- $\mathsf{Sign}(\mathsf{usk}_{\mathbb{A}}, m, \Upsilon)$: outputs a signature σ where $\Upsilon(\mathbb{A}) = 1$
- $\mathsf{Verify}(m, \sigma, \mathsf{mpk}, \Upsilon)$: outputs 1 if σ is valid, 0 otherwise.

Security properties

The perfect privacy is a security requirement for ABS. The signer's privacy relies only on the signature trustee and not the authority.

An ABS achieves perfect privacy if the two following distributions are equal:

$$\mathsf{Sign}(\mathsf{param}, \mathsf{sk}_1, m, \Upsilon) = \mathsf{Sign}(\mathsf{param}, \mathsf{sk}_2, m, \Upsilon)$$

with: $\mathsf{sk}_i \leftarrow \mathsf{KeyGen}(\mathsf{param}, \mathbb{A}_i)$
and $\Upsilon(\mathbb{A}_1) = \Upsilon(\mathbb{A}_2) = 1$.

3 Building Blocks

In this section, we now give explicit instantiation of the various tools we need.

3.1 Downgradable IBE

We first present a downgradable IBE (DIBE) from Blazy et al. (2019), such schemes allow each user with a valid key for an identity id, to compute a valid key for every $\tilde{\mathsf{id}} \preceq \mathsf{id}$ where $\preceq$ is a predetermined relation.[1]

Definition 6 A Downgradable IBE (DIBE) consists of five probabilistic polynomial time (ppt) algorithms (Setup, USKGen, Encap, Decap, USKDown):

- $\mathsf{Setup}(\mathfrak{K})$: outputs the master public key mpk and the master secret key msk.
- $\mathsf{USKGen}(\mathsf{msk}, \mathsf{id})$: outputs the user secret key associated to his identity id usk[id] and a delegation key udk[id].
- $\mathsf{Encap}(\mathsf{mpk}, \mathsf{id})$: outputs a ciphertext c and a symmetric key K.
- $\mathsf{USKDown}(\mathsf{usk}[\mathsf{id}], \tilde{\mathsf{id}})$ return the user secret key $\mathsf{usk}[\tilde{\mathsf{id}}]$ as long as $\tilde{\mathsf{id}} \preceq \mathsf{id}$.
- $\mathsf{Decap}(\mathsf{usk}[\mathsf{id}], \mathsf{id}, c)$: outputs the decapsulated key K or $\perp$.

We recall the DIBE construction based on the tight IBE from Blazy et al. (2014) and PR-ID-CPA secured under the $\mathcal{D}_k$-MDDH assumption in appendix C. We use this construction in our instantiation (Figure 2).

Following the same idea, one can define an *Upgradable* IBE, with an algorithm USKUp which upgrades a fresh secret key as long as $\mathsf{id} \preceq \tilde{\mathsf{id}}$. The generated key, noted $\mathsf{uusk}[\tilde{\mathsf{id}}]$ allows the verification of the signature if and only if verifier's attributes match with the attributes needed to verify. For practical use, we consider only the presence of one of this algorithms. In presence of this two public algorithms, each user could generate any key.

Thanks to previous definitions, we are able to present our Attribute-Based Designated Verifier Signature.

3.2 Attribute-Based Designated Verifier Signature

An Attribute-Based Designated Verifier Signature, noted ABDVS, uses the different blocks presented earlier in order to verify a signature with only some attributes.

The Figure 1 highlights the use of these different blocks.

- Thanks to the Downgradable IBE we generate a key for $\mathsf{id}_1 || m$ by downgrading the key received by the user id_1, using the Naor transform, this is an equivalent of id-signing m.
- DIBE protocol generates verifier's keys and allows keys upgrade. It also used to mask the signature.
- The USKUp algorithm allows designated verifier to obtain his key uusk, generated by all his attributes, by upgrading his old key usk.

Concerning the security requirements, ABDVS has to verify unforgeability, non-transferability and perfect privacy. In the following, we are showing that properties can be proved by relying on the security of protocols under used.

4 Our construction

4.1 Framework

The concept of Attribute-Based Encryption (ABE) was introduced by Sahai and Waters in Sahai and Waters (2005). In an ABE, a key is associated with a set $\mathbb{A}$ of attributes while a ciphertext is associated with an access policy[2] $\mathbb{F}$. The decryption can be done if $\mathbb{A}$ satisfies $\mathbb{F}$. As in Blazy et al. (2019), we work with a small-universe of attributes noted $\mathcal{U} = \{1, \ldots, \ell\}$, with ℓ the length of the identity used in our construction. For any set $S \subseteq \mathcal{U}$, we define $\mathsf{id}_S \in \{0,1\}^\ell$ where its i-th position is defined by:

$$\mathsf{id}_S[i] := \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{else} \end{cases} \tag{1}$$

In our construction, we used boolean formula in Disjunctive Normal Form (DNF). We note by k the number of disjunction in the DNF formula.

Our construction is presented in Figure 1.

[1] The relation leads to a partial order, *i.e.* $\forall x, y, x \neq y \Rightarrow x \preceq y = \mathtt{false} \vee y \preceq x = \mathtt{false}$.

[2] For simplicity we assume that $\mathbb{F}$ is a policy expressed in DNF form

$\underline{\mathsf{Setup}(\mathfrak{K})}$:
$\mathsf{mpk}, \mathsf{msk} = \mathsf{DIBE.Setup}(\mathfrak{K})$
Return $(\mathsf{mpk}, \mathsf{msk})$

$\underline{\mathsf{KeyGen}_{\mathsf{Sign}}(\mathsf{msk}, \mathsf{id}_1)}$:
$\mathsf{sk}, \mathsf{ek} \leftarrow \mathsf{DIBE.USKGen}(\mathsf{msk})$
Return sk, ek

$\underline{\mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, \mathsf{id}_2)}$:
$\mathsf{usk}, \mathsf{dk} \leftarrow \mathsf{DIBE.USKGen}(\mathsf{msk})$
Return $\mathsf{usk}, \mathsf{dk}$

$\underline{\mathsf{Sign}(\mathsf{mpk}, (\mathsf{usk}, \mathsf{ek}, \mathsf{id}_1), m, \mathbb{F})}$:
$\sigma_1, \sigma_2 \leftarrow \mathsf{DIBE.USKGen}(\mathsf{usk}, \mathsf{id}_1 || m)$
Parse $\mathbb{F} = \bigvee_{j=1}^{k} (\bigwedge_{i, \mathsf{id}_{j,i}=1} 1)$
For all $j \in [1, k]$, compute:
$(c_j, K_j) \leftarrow \mathsf{DIBE.Encap}(\mathsf{mpk}, \mathsf{id}_j, \sigma_2)$
Return $C = (c_1, ..., c_k)$, $\mathsf{K} = (\mathsf{K}_1, ..., \mathsf{K}_j)$ and σ_1

$\underline{\mathsf{Verify}(\mathsf{sk}_2, C)}$:
Parse $\mathbb{F} = \bigvee_{j=1}^{k} (\bigwedge_{a \in I} a)$
Find $j \in [1, k]$ s.t. $S_j \subseteq \mathbb{A}$
$\mathsf{uusk} \leftarrow \mathsf{USKUp}(\mathsf{usk}, \mathsf{id}_{2,S_j})$
$K_j \leftarrow \mathsf{DIBE.Decap}(\mathsf{uusk}, \mathsf{id}_{2,S_j}, C_j)$

Fig. 1 Generic algorithms of our ABDVS framework

At first, each potential verifier asks the authority to generate a set of k secret keys corresponding to his policy where k is the number of disjunctions on his policy.

When he wants to verify a signature, he selects one of his key, that can be derived into the expected verification key, noted uusk using the USKUp algorithm.

Then, thanks to a pairing verification, the designated verifier checks the validity of the signature only if his policy matches with attributes needed for the verification. An important point is if none of the key works, the verifier cannot gather any information.

Now, we present a concrete instantiation of our framework in Figure 2.

We note by id_1 the identity containing the attributes needed for the verification. id_2 is the identity of the verifier *i.e.* id_2 contains all the verifier's attributes. In Figure 2, we use two KeyGen algorithms:

- $\mathsf{KeyGen}_{\mathsf{Sign}}$: generates the signing key sk and ek used for signature generation.
- KeyGen: generates the verifier user secret key usk for id_2 and dk used to randomize usk.

Thanks to encapsulation of the DIBE protocol, we hide our signature with K. The capsule, C, allows signature verification in Decap algorithm.

In USKUp, we modify usk in uusk. The verifier, with uusk, is able to verify the signature because uusk contains attributes required.

Decap verify the signature thanks to the decapsulation of C.

4.2 Security Aspect

Concerning the security, our ABDVS has to verify three security properties: unforgeability (theorem 1), non-transferability (theorem 2) and perfect privacy (theorem 3).

For our construction, we don't have to prove DV-unforgeability because this property is similar to unforgeability.

First, we show the correctness of our protocol. In other words, a signature generated honestly can be verified by an honest verifier.

The signer computes: $C = [(c_{j,0}, c_{j,1})_j]_2$, $\mathsf{K} = [K_j]_T$, $\sigma_1 = ([s']_1, [s']_2)$.

By simplifying the first pairing in the verification, we get:

$$[c_0 \mathsf{uusk}_2 - c_1 \mathsf{uusk}_1]_T = [\mathsf{K} + \sigma_1 \sum_{i=0}^{\ell} \tilde{\mathsf{id}}_i z_i \sum_{i=\ell+1}^{2\ell} m_{i-\ell} z_i]_T$$

The previous equation is obtained by using that, in the honest case, $\mathsf{uusk}_2 = \sum_{i=1}^{l} \mathsf{id}_i z_i$ which is in c_1. We can see that this equality is verified as soon as the attributes used by the verifier are the ones expected by the signer. Using bilinearity, we can see that $v' = az' + u \sum_{i=0}^{2\ell} \tilde{id}_i z_i = az' + \sigma_1 \sum_{i=0}^{\ell} \tilde{\mathsf{id}}_i z_i + \sum_{i=\ell+1}^{2\ell} m_{i-\ell} z_i$.

Theorem 1 *Under the indistiguishability of the* DIBE *used, our construction is unforgeable under the SXDH assumption.*

Proof As presented in the article Cui et al. (2007) by Cui *et al.*, a signature generated thanks to the Naor transformation, is unforgeable if the IBE used is semantically secure. The IBE used in our construction is IND-ID-CPA under the BDH assumption. So, our ABDVS is unforgeable under the SXDH assumption.

Theorem 2 *Our* ABDVS *is non-transferable under the* IND-ID-CPA *security of the* IBE.

Proof Let $\mathcal{A}$ be an adversary against the non-transferability of our signature. In this proof, we are going to build a simulator $\mathcal{B}$ giving $\mathcal{A}$ access to USKGen, USKUp and Encap, and using $\mathcal{A}$ to break the IND-ID-CPA security of the IBE scheme.

$\underline{\mathsf{Setup}(\mathfrak{K})}$:
For $i \in \{1, 2\ell\} : z_i \xleftarrow{\$} \mathbb{Z}_p$;
$a, z' \xleftarrow{\$} \mathbb{Z}_p; y' = az'$;
$\mathsf{mpk} = (([z_i]_{1,2})_{1 \leq i \leq 2\ell}, [a]_2, [z']_{1,2})$
$\mathsf{msk} = ((z_i)_{1 \leq i \leq 2\ell}), y')$
Return $(\mathsf{mpk}, \mathsf{msk})$

$\underline{\mathsf{KeyGen}_{\mathsf{Sign}}(\mathsf{msk}, \mathsf{mpk}, \mathsf{id}_1, \perp) :}$
$s \xleftarrow{\$} \mathbb{Z}_p$
$v = y' + (\sum_1^\ell \mathsf{id}_{1,i} z_i)s$
For $i \in [\ell + 1, 2\ell]$:
  $e_i = z_i \cdot s \in \mathbb{Z}_p$;
$\mathsf{sk} = ([s]_1, [s]_2, [v]_1)$
Return $(\mathsf{sk}, \mathsf{ek} = [e_i]_1)$

$\underline{\mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, \mathsf{id}_2, \perp) :}$
$s \xleftarrow{\$} \mathbb{Z}_p$
$v = y' + (\sum_1^\ell \mathsf{id}_{2,i} z_i)s$
For $i, \mathsf{id}[i] = 1$:
  $d_i = z_i \cdot s \in \mathbb{Z}_p$;
$\mathsf{usk} = ([s]_1, [s]_2, [v]_1)$
Return $(\mathsf{usk}, \mathsf{dk} = [d_i]_1)$

$\underline{\mathsf{Sign}(\mathsf{mpk}, (\mathsf{usk}, \mathsf{ek}, \mathsf{id}_1), m, \mathbb{F}) :}$
$u \xleftarrow{\$} \mathbb{Z}_p$, $s' = s + u$
$v' = v + (\sum_{i=\ell+1}^{2\ell} m_{i-\ell} z_i)s' + (\sum_{i=0}^{\ell} \mathsf{id}_{1,i} z_i)u$
$\sigma = (\sigma_{1,1}, \sigma_{1,2}, \sigma_2) = ([s']_1, [s']_2, [v']_1)$
Parse $\mathbb{F} = \bigvee_{j=1}^{k} \bigwedge_{\mathsf{id}_{j,i}=1} 1$
For all $j \in [1, k]$, compute:
  $r_j \xleftarrow{\$} \mathbb{Z}_p$
  $c_{j,0} = ar_j$
  $c_{j,1} = (\sum_{i=1}^{\ell} \mathsf{id}_{j,i} z_i) \cdot r_j$
  $K_j = az' \cdot r_j + \sigma_2 \in \mathbb{Z}_p$
Return $C = [(c_{j,0}, c_{j,1})_j]_2$, $\mathsf{K} = [K_j]_T$, $\sigma_1 = ([s']_1, [s']_2)$.

$\underline{\mathsf{USKUp}(\mathsf{usk}[\mathsf{id}_2], \tilde{\mathsf{id}}, \mathsf{dk}):}$
If $\neg(\mathsf{id}_2 \preceq \tilde{\mathsf{id}})$, then return $\perp$.
Set $\mathcal{I} = \{i | \tilde{\mathsf{id}}[i] = 1 \wedge \mathsf{id}_2[i] = 0\}$
Upgrading the key:
  $\hat{v} = v + \sum_{i \in I} \mathsf{id}_{2,S_j,i} d_i \in \mathbb{Z}_p$
$\mathsf{uusk}[\tilde{\mathsf{id}}] = ([s]_1, [\hat{v}]_1)$
Return $\mathsf{uusk}[\tilde{\mathsf{id}}]$

$\underline{\mathsf{Verify}(\mathsf{mpk}, \mathsf{uusk}[\tilde{\mathsf{id}}], m, C, \mathsf{K}, \sigma_1):}$
Parse $\mathsf{uusk}[\tilde{\mathsf{id}}] = ([s]_1, [\hat{v}]_1)$.
Find a j, where $\mathsf{id}_j \in \mathbb{F} \wedge \mathsf{id} \preceq \mathsf{id}_j$
From C, K, extract $([c_{j,0}]_2, [c_{j,1}]_2), [K_j]_T$.
Check whether:
$[c_0 \mathsf{uusk}_2 - c_1 \mathsf{uusk}_1]_T =$
$[\mathsf{K} - (\sigma_{1,2}(\sum_{i=0}^{\ell} \mathsf{id}_i z_i + \sum_{i=\ell+1}^{2\ell} m_{i-\ell} z_i))]_T$
and $[\sigma_{1,1}]_T = [\sigma_{1,2}]_T$

Fig. 2 Algorithms of our ABDVS scheme based on the SXDH assumption

$\underline{\mathsf{Sign}(\mathsf{id}^*) :}$ G_0, $\boxed{G_1}$
Parse $\mathbb{F} = \bigvee_{j=1}^{k} \mathsf{id}^{(j)}$
For all $j \in [1, k]$, compute:
  $r_j \xleftarrow{\$} \mathbb{Z}_p$
  $\alpha \xleftarrow{\$} \mathbb{Z}_p$
  $c_{j,0} = a \cdot r_j \in \mathbb{Z}_p$
  $c_{j,1} = (\sum_{i=1}^{\ell} \mathsf{id}_i z_i) \cdot r_j$
  $K_j = z' \cdot r_j + \sigma_2$
  $\boxed{K_j = z' \cdot r_j + \alpha}$
  Return $C = [(c_{j,0}, c_{j,1})_j]_2$ and $K = [K_j]_T$.

Fig. 3 Security games $\mathbf{G}_0$-$\mathbf{G}_1$ for the non-transferability

Game 0. This is the real attack game. We don't modify existing algorithms.

Game 1. $\mathcal{A}$ queries m times USKGen and Encap for different identities but not the challenged one. The simulator answers honestly to each query.

For the identity challenge, id^*, we replace the signature σ_2 by a random group element α. This modification in the algorithm Encap is presented in Figure 3. In other words, we randomize a part of the key K_j. The indistinguishability of a real key from a random one is ensured by the indistinguishability of the DIBE used in our construction. Hence, $\mathbf{G_1}$ is indistinguishable from $\mathbf{G_0}$. Thus, we have:

$$\mathsf{Adv}^{\mathbf{G_0},\mathbf{G_1}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{ANON\text{-}ID\text{-}CPA}}_{\mathsf{IBE}}(\mathcal{A}).$$

Theorem 3 *Our signature achieves perfect privacy under the* ANON-ID-CPA *security of the* DIBE.

Proof Let $\mathcal{A}$ be an adversary against the perfect privacy of our signature. In this proof, we are going to build a simulator $\mathcal{B}$ giving $\mathcal{A}$ access to USKGen, USKUp and Encap, and using $\mathcal{A}$ to break the ANON-ID-CPA security of the IBE.

Game 0. This is the real attack game. We don't modify existing algorithms.

Game 1. In this game, $\mathcal{B}$ simulates the signing key associated to different attributes sets. This simulation is presented in Figure 4 where α is either one of the attribute set. The difference between the two signing procedures can directly be reduced to the difference between the output of the Encap algorithm associated with the two different possible id_α, hence the ANON-ID-CPA property of the underlying DIBE.

If he is able to figure out the difference, he breaks the anonymity of the DIBE used. Thus, we have:

$\mathsf{Adv}^{\mathbf{G_0},\mathbf{G_1}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{ANON\text{-}ID\text{-}CPA}}_{\mathsf{DIBE}}(\mathcal{A})$.

In the following section, we present computational times of our protocol.

5 Performance evaluation

In this section we will first start with a theoretical comparison of our work with state-of-the-art where we will focus on communication costs based on cryptographic keys sizes and signatures sizes. We then present a practical evaluation of an implementation of our proposition and an implementation of Fan *et al.* work.

5.1 Theoretical comparison

In Table 2, we compare our solutions to other existing schemes. It must be noted that only Fan *et al.* offers the same security properties as our proposition as depicted in Table 1. Two other schemes are presented in this comparison to have some ground truth for both key and signatures sizes. In all schemes, elements in the target group are evaluated as hashed bitstring in order to gain efficiency.

We inherit from Attribute-Based construction, the linear size of the public key in term of the number n of attributes. We can see, that compared to the other existing Attribute-Based Designated Verifier Signature scheme, we drop the dependency in the number u of users in the signature, we are still linear in the number k of disjunctions in the policy. In term of security hypotheses, our scheme is proven in the standard model under a classical hypothesis, which is also an upgrade compared to the reliance on the Random Oracle Model in the previous scheme.

In Table 3 is depicted the time taken by the three main algorithm of our two schemes (e.g. Setup,Sign and Verify). t_inv is the time taken to do an inversion; t_m is the time taken to perform a multiplication between two group elements; t_e is the time to do a modular exponentiation between a group element and a scalar in $\mathbb{Z}_p$; t_p is the time to realize a pairing operation between two group elements; t_s the time cost of a scalar multiplication in $\mathbb{G}_1$. As a reminder, ℓ is the number of attributes and k the number of disjunctions under use. Based on these different operations, Table 3 shows that our proposition is expected to have better results than Fan *et al.* for both Sign and Verify as we have less complexity on these operations. Please note, that compared to the table in the initial paper, we consider their scheme have a linear dependency in the policy size, as their Verify algorithm is expected to do a computation for every node in the tree they build. Fan *et al.* have better results for Setup as they only rely on inversions and multiplication between two group elements whereas we need some modular exponentiation which is more time consuming. But, this is not problematic for real world applications as this step is performed only once prior to communication between parties (i.e offline).

5.2 Practical evaluation

The aim of this prototype and the proposed experiment is to highlight the usability of the framework for real world applications in terms of computation time. To achieve that, we use a virtual machine with 4 GB of dedicated RAM running on a host computer with a 2.6 GHz Intel Core i7 processor. Since Fan et al. proposed the only attribute-based designated verifier scheme to our knowledge, we use it to compare our performances.

Our scheme is implemented with Python Library Charm-Crypto and the Asymmetric Charm curve *MNT159* while Fan et al. scheme is implemented using the Symmetric Charm Curve *SS512*. It was not possible to use the same curve as the two schemes are not using the same kind of pairings (asymmetric against symmetric). We obtain better performances for our scheme for two reasons, we can rely on much faster curves, and due to the way we handle policy, we have no efficiency loss when increasing the number of disjunctions during verification.

In all experiments, the signing operation is performed on a hashed message with a length of 128bits. All computation times are displayed in Table 4 and were computed for ten attributes.

It can be viewed in Table 4 that our scheme takes less time to generate master and secret keys than Fan et al. (2012). Indeed, our KeyGen algorithm depends only on the number of attributes as for Fan et al. proposal but our key sizes are smaller which explains this better performances. It must also be noted that in our evaluation, KeyGen algorithm generates both verify and signing key. As previously explained, we can also see that the Verify algorithm has much better performances. Indeed, with our design, users can pinpoint the disjunctions their credential fulfill when checking the validity of a signature avoiding a linear loss, which allows us to have a constant verification time.

As illustrated in Figure 5 and Figure 6, experimental results for Sign and Verify algorithms are more efficient and outperforms Fan et al. solution. Their signing algorithm depends on the number of disjunctions as ours do, but we require fewer operations. However, we are only using a constant number of pairings in the verification algorithm.

$\mathsf{KeyGen}_{\mathsf{Sign}}(\mathsf{msk}, \mathsf{mpk}, \mathsf{id}, \perp)$:

$s \xleftarrow{\$} \mathbb{Z}_p$
$v = y' + (\sum_1^\ell \mathsf{id}_{1,i} z_i)s$
For $i \in [\ell + 1, 2\ell]$:
  $e_i = z_i \cdot s \in \mathbb{Z}_p$;

$\mathsf{usk} := ([s]_1, [s]_2, [v]_1)$
  Return $(\mathsf{usk}, \mathsf{ek} = [e_i]_1)$

$\mathsf{Sign}_\alpha(\mathsf{mpk}, \mathsf{id}, \mathsf{usk}, \mathsf{ek}, m, \mathbb{F})$:

Parse $\mathbb{F} = \bigvee_{j=1}^k \mathsf{id}_\alpha^{(j)}$
For all $j \in [1, k]$, compute:
  $r_j \xleftarrow{\$} \mathbb{Z}_p$
  $c_{j,0} = a \cdot r_j \in \mathbb{Z}_p$
  $c_{j,1} = (\sum_{i=1}^\ell \mathsf{id}_{\alpha,i} z_i) \cdot r_j$
  $K_j = z' \cdot r_j + \sigma_2$
Return $C = [(c_{j,0}, c_{j,1})_j]_1$ and $K = [K_j]_T$.

Fig. 4 Security games $\mathbf{G}_0$-$\mathbf{G}_1$ for the perfect privacy

	Type	PK Size	Sign. Size	Hypothesis
Katsumata et al. (2020)	ABS	$(4n+1) \cdot \mathbb{G}_1$	$2 \cdot \mathbb{G}_2$	SXDH
Laguillaumie and Vergnaud (2004)	DVS	$1 \cdot \mathbb{G}_1$	$2 \cdot \mathbb{Z}_p$	DBDH
Fan et al. (2012)	ABDVS	$2n \cdot \mathbb{Z}_p$	$3k \cdot \mathbb{G} + 2u \cdot \mathbb{G} + 5 \cdot \mathbb{Z}_p$	DBDH+ROM
Our ABDVS	ABDVS	$n \cdot \mathbb{G}_1$	$3k \cdot \mathbb{G}_1 + k \cdot \mathbb{Z}_p$	SXDH

Table 2 Comparison of various schemes.

	Fan *et al.*	Our ABDVS
Setup	$3\ell \cdot t_{inv} + 9\ell \cdot t_m$	$(2\ell + 1) \cdot t_e$
Sign	$5 \cdot t_p + 1 \cdot t_{inv} + (k + 7\ell) \cdot t_m$	$3\ell \cdot t_e + k \cdot (\ell + 1) \cdot t_m + \ell \cdot t_p$
Verify	$5 \cdot t_p + 1 \cdot t_{inv} + 5 \cdot t_s + 5\ell \cdot t_m$	$4 \cdot t_p + \ell \cdot t_m$

Table 3 Comparison with Fan *et al.* in terms of number of operations.

Table 4 Numerical evaluation of ABDVS

Algorithm	Fan et al. (2012)	Our Proposal
Setup	22 *ms*	62 *ms*
KeyGen	71 *ms*	34 *ms*
USKUp	$\emptyset$	< 1 *ms*
Sign	63 *ms*	22 *ms*
Verify	207 *ms*	7 *ms*

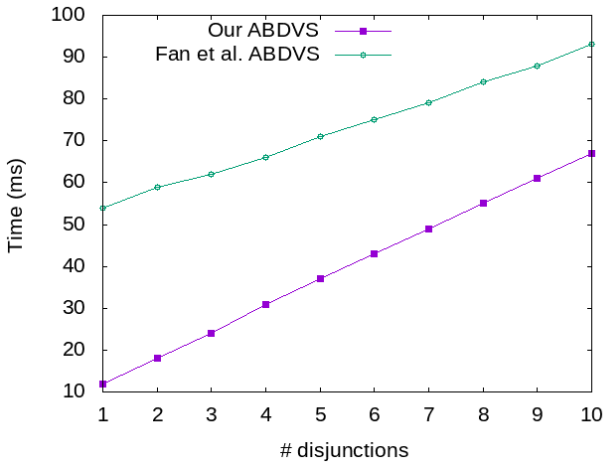

Fig. 5 Signing with Fan et al. (2012) and our ABDVS

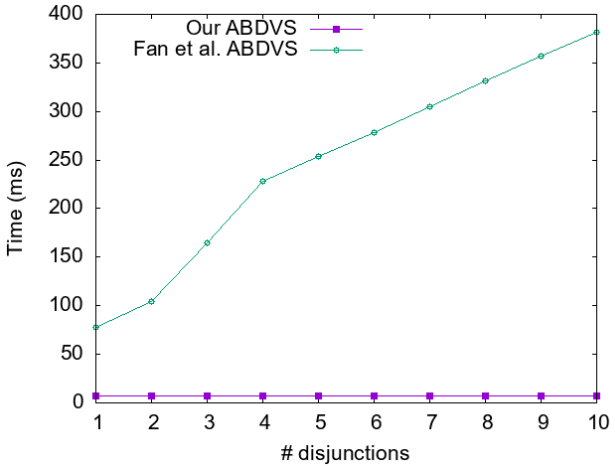

Fig. 6 Verifying with Fan et al. (2012) and our ABDVS

Practical instantiations would also be concerned with batching verification of several signatures. This concept was introduced in Fiat (1990). In 1998, Bellare et al. (1998) proposed the first systematic look at batch verification and described several techniques for conducting batch verification of exponentiations with high confidence. More recently Ferrara et al. (2009) presented a careful study on how to securely batch-verify a set of pairing-based equations. As our signature scheme verification only requires evaluation of pairing product equations, these techniques can also be applied to drastically reduce the number of pairings evaluation by transforming most of the additional ones into exponentiation and group elements products.

6 Conclusion

In this paper we propose a new construction of attribute-based designated verifier signature in the standard model. This construction relies on Blazy-Kiltz-Pan IBE and a transformation to obtain ABE from DIBE. Compare to the previous existing proposition, this construction provides better performances to generate a designated verifier signature and to verify it. It makes it suitable for real case usage such as in a cloud context where it can be used to perform anonymous authentication between users and a cloud server. Furthermore the proposed construction can be generalized so that it is possible to designate attributes for verification instead of an identity which opens new perspectives for cloud environments and health applications where attribute-based encryption is widely used.

One current limitation of our scheme is that the generated encapsulation is linear in the size k of the policy. One may hope, that by using standard packing technique, one can obtain a more reasonable size in $\log(k)$ or even achieve a constant size. Another line of research, would be to prepare for the advent of quantum computing, and as such future work might focus on proposing a lattice-based version of such scheme. Lattices have the basic tool required to build an ABDVS, some extra care would however be needed to take care of the noise growth and as such a naive version with a bounded policy size might be an interesting first step.

References

Ateniese G, Camenisch J, Hohenberger S, de Medeiros B (2005) Practical group signatures without random oracles. Cryptology ePrint Archive, Report 2005/385, http://eprint.iacr.org/2005/385

Bellare M, Garay JA, Rabin T (1998) Fast batch verification for modular exponentiation and digital signatures. In: Nyberg K (ed) EUROCRYPT'98, Springer, Heidelberg, LNCS, vol 1403, pp 236–250

Blazy O, Kiltz E, Pan J (2014) (Hierarchical) identity-based encryption from affine message authentication. In: Garay JA, Gennaro R (eds) CRYPTO 2014, Part I, Springer, Heidelberg, LNCS, vol 8616, pp 408–425, DOI 10.1007/978-3-662-44371-2_23

Blazy O, Conchon E, Germouty P, Jambert A (2017) Efficient id-based designated verifier signature. In: Proceedings of the 12th International Conference on Availability, Reliability and Security, Reggio Calabria, Italy, August 29 - September 01, 2017, ACM, pp 44:1–44:8, DOI 10.1145/3098954.3103157, URL https://doi.org/10.1145/3098954.3103157

Blazy O, Germouty P, Phan DH (2019) Downgradable identity-based encryption and applications. In: Topics in Cryptology - CT-RSA 2019 - The Cryptographers' Track at the RSA Conference 2019, San Francisco, CA, USA, March 4-8, 2019, Proceedings, pp 44–61, DOI 10.1007/978-3-030-12612-4_3, URL https://doi.org/10.1007/978-3-030-12612-4_3

Chaum D, Van Antwerpen H (1989) Undeniable signatures. In: Conference on the Theory and Application of Cryptology, Springer, pp 212–216

Cui Y, Fujisaki E, Hanaoka G, Imai H, Zhang R (2007) Formal security treatments for signatures from identity-based encryption. In: Susilo W, Liu JK, Mu Y (eds) ProvSec 2007, Springer, Heidelberg, LNCS, vol 4784, pp 218–227

Escala A, Herold G, Kiltz E, Ràfols C, Villar J (2013) An algebraic framework for Diffie-Hellman assumptions. In: Canetti R, Garay JA (eds) CRYPTO 2013, Part II, Springer, Heidelberg, LNCS, vol 8043, pp 129–147, DOI 10.1007/978-3-642-40084-1_8

Fan CI, Wu CN, Chen WK, Sun WZ (2012) Attribute-based strong designated-verifier signature scheme. J Systems and Software 85:944–959

Ferrara AL, Green M, Hohenberger S, Pedersen MØ (2009) Practical short signature batch verification. In: Fischlin M (ed) CT-RSA 2009, Springer, Heidelberg, LNCS, vol 5473, pp 309–324

Fiat A (1990) Batch RSA. In: Brassard G (ed) CRYPTO'89, Springer, Heidelberg, LNCS, vol 435, pp 175–185

Goldwasser S, Micali S, Rivest RL (1988) A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal on Computing 17(2):281–308

Huang X, Susilo W, Mu Y, Wu W (2008) Secure universal designated verifier signature without random oracles. Int J Inf Sec 7(3):171–183, DOI 10.1007/s10207-007-0021-2, URL https://doi.org/10.1007/s10207-007-0021-2

Jakobsson M, Sako K, Impagliazzo R (1996) Designated verifier proofs and their applications. In: Maurer UM (ed) EUROCRYPT'96, Springer, Heidelberg, LNCS, vol 1070, pp 143–154

Katsumata S, Nishimaki R, Yamada S, Yamakawa T (2020) Compact nizks from standard assumptions on bilinear maps. In: Canteaut A, Ishai Y (eds) Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Tech-

niques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part III, Springer, Lecture Notes in Computer Science, vol 12107, pp 379–409, DOI 10.1007/978-3-030-45727-3_13, URL https://doi.org/10.1007/978-3-030-45727-3_13

Laguillaumie F, Vergnaud D (2004) Designated verifier signatures: Anonymity and efficient construction from any bilinear map. In: Blundo C, Cimato S (eds) Security in Communication Networks, 4th International Conference, SCN 2004, Amalfi, Italy, September 8-10, 2004, Revised Selected Papers, Springer, Lecture Notes in Computer Science, vol 3352, pp 105–119, DOI 10.1007/978-3-540-30598-9_8, URL https://doi.org/10.1007/978-3-540-30598-9_8

Laguillaumie F, Libert B, Quisquater J (2006) Universal designated verifier signatures without random oracles or non-black box assumptions. In: Prisco RD, Yung M (eds) Security and Cryptography for Networks, 5th International Conference, SCN 2006, Maiori, Italy, September 6-8, 2006, Proceedings, Springer, Lecture Notes in Computer Science, vol 4116, pp 63–77, DOI 10.1007/11832072_5, URL https://doi.org/10.1007/11832072_5

Maji HK, Prabhakaran M, Rosulek M (2011) Attribute-based signatures. In: Kiayias A (ed) CT-RSA 2011, Springer, Heidelberg, LNCS, vol 6558, pp 376–392

Sahai A, Waters BR (2005) Fuzzy identity-based encryption. In: Cramer R (ed) EUROCRYPT 2005, Springer, Heidelberg, LNCS, vol 3494, pp 457–473

Steinfeld R, Bull L, Wang H, Pieprzyk J (2003) Universal designated-verifier signatures. In: Laih CS (ed) ASIACRYPT 2003, Springer, Heidelberg, LNCS, vol 2894, pp 523–542, DOI 10.1007/978-3-540-40061-5_33

Susilo W, Zhang F, Mu Y (2004) Identity-based strong designated verifier signature schemes. In: Australasian Conference on Information Security and Privacy, Springer, pp 313–324

A Digital Signature

Definition 7 A signature scheme is composed by four algorithms:

- $\mathsf{Setup}(\mathfrak{K})$: generates the global parameter of the system param.
- $\mathsf{KeyGen}(\mathsf{param})$: outputs a pair of key $(\mathsf{sk}, \mathsf{pk})$ where sk is the (secret) signing key and pk the (public) verification key.
- $\mathsf{Sign}(\mathsf{sk}, m; \mu)$: outputs a signature σ on the message m thanks under sk, and some randomness μ.
- $\mathsf{Verify}(\mathsf{vk}, m, \sigma)$: checks the validity of the signature σ with vk.

Digital Signature has to verify two security properties: correctness and existential unforgeability under chosen message attacks ($\mathsf{EUF-CMA}$).

- Correctness: For every pair $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(\mathsf{param})$, for every message $m \in \mathcal{M}$ and for all randomness μ, we have $\mathsf{Verify}(\mathsf{vk}, m, \mathsf{Sign}(\mathsf{sk}, m; \mu)) = 1$.
- *Existential Unforgeability under Chosen Message Attacks Goldwasser et al. (1988)*: even after querying n valid signatures on chosen messages (m_i), $\mathcal{A}$ should not be able to output a valid signature on a fresh message m. We define a signing oracle:
  $\mathsf{OSign}(\mathsf{vk}, m)$: outputs a signature on m valid under the verification key vk. The requested message is added to the signed messages set $\mathcal{SM}$.

$\mathsf{Exp}^{\mathsf{euf}}_{\mathcal{S},\mathcal{A}}(\mathfrak{K})$
1. $\mathsf{param} \leftarrow \mathsf{Setup}(1^{\mathfrak{K}})$
2. $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{param})$
3. $(m^*, \sigma^*) \leftarrow \mathcal{A}(\mathsf{vk}, \mathsf{OSign}(\mathsf{vk}, \cdot))$
4. $b \leftarrow \mathsf{Verify}(\mathsf{vk}, m^*, \sigma^*)$
5. `IF` $m^* \in \mathcal{SM}$ `RETURN` 0
6. `ELSE RETURN` b

Fig. 7 $\mathsf{EUF-CMA}$ Game for a Signature Scheme

The probability of success against the game given in Figure 7 is denoted by

$$\mathsf{Succ}^{\mathsf{euf}}_{\mathcal{S},\mathcal{A}}(\mathfrak{K}) = \Pr[\mathsf{Exp}^{\mathsf{euf}}_{\mathcal{S},\mathcal{A}}(\mathfrak{K}) = 1],$$
$$\mathsf{Succ}^{\mathsf{euf}}_{\mathcal{S}}(\mathfrak{K}, t) = \max_{\mathcal{A} \leq t} \mathsf{Succ}^{\mathsf{euf}}_{\mathcal{S},\mathcal{A}}(\mathfrak{K}).$$

B Designated Verifier Signature: Security Properties

A DVS has to verify different security properties:

- Unforgeability as any regular signature. Even after querying n valid signatures on chosen messages, an adversary should not be able to output a valid signature on a fresh message.
- DV-unforgeability: only the signer or the designated verifier should be able to generate a verifiable message signature pair for (m,$\hat{\sigma}$). The security experiment is presented in Figure 8.

$\mathsf{Exp}^{\mathsf{UDVS}}_{\mathsf{dvuf},\mathcal{A}}(\mathfrak{K})$
1. $(\mathsf{param}) \leftarrow \mathsf{Setup}(\mathfrak{K})$
2. $(\mathsf{sk}_1, \mathsf{pk}_1) \leftarrow \mathsf{KGS}(\mathsf{param})$
3. $(\mathsf{sk}_2, \mathsf{pk}_2) \leftarrow \mathsf{KGV}(\mathsf{param})$
4. $(m, \hat{\sigma}) \leftarrow \mathcal{A}^{\mathsf{OSign},\mathsf{OVerify}}(\mathsf{pk}_1, \mathsf{pk}_2)$;
6. `RETURN` $\mathsf{DesV}(\mathsf{sk}_2, \mathsf{pk}_1, m, \hat{\sigma})$ if $m \notin \mathcal{SM}$

Fig. 8 Unforgeability Experiment for DVS

We denote by KGS: Key Generation Signature and KGV: Key Generation Verification. In Figure 8, we used two oracles described below.

- $\mathsf{OSign}(\mathsf{pk}_1, m)$: outputs a signature σ on the message m and adds m to the set of signed messages $\mathcal{SM}$.
- $\mathsf{OVerify}(\mathsf{pk}_2, m, \hat{\sigma})$: checks the validity of the designated signature $\hat{\sigma}$.

– **Non-transferability**: an adversary should not be able to convince a third party about the validity (or invalidity) of a designated signature. An adversary $\mathcal{A}$ must have at best a negligible advantage in distinguishing the two following distributions:

$$\Delta_0 = \left\{ (m, \hat{\sigma}) \middle| \begin{array}{l} (\mathsf{sk}_1, \mathsf{pk}_1) \leftarrow \mathsf{KGS}(\mathsf{param}) \\ (\mathsf{sk}_2, \mathsf{pk}_2) \leftarrow \mathsf{KGV}(\mathsf{param}) \\ \hat{\sigma} = \mathsf{Des}(\mathsf{pk}_1, \mathsf{pk}_2, m, \mathsf{Sign}(\mathsf{sk}_1, m)) \end{array} \right\}$$

$$\Delta_1 = \left\{ (m, \hat{\sigma}) \middle| \begin{array}{l} (\mathsf{sk}_1, \mathsf{pk}_1) \leftarrow \mathsf{KGS}(\mathsf{param}) \\ (\mathsf{sk}_2, \mathsf{pk}_2) \leftarrow \mathsf{KGV}(\mathsf{param}) \\ \hat{\sigma} \leftarrow \mathcal{S} \end{array} \right\}$$

C Downgradable IBE

We present in Figure 9 the DIBE used in our protocol.

Theorem 4 *Under the $\mathcal{D}_k$-MDDH assumption, the* DIBE *is* PR-ID-CPA *secure. For all adversaries $\mathcal{A}$, there exists an adversary $\mathcal{B}$ with* $\mathsf{TIME}(\mathcal{A}) \approx \mathsf{TIME}(\mathcal{B})$ *and*

$$\mathsf{Adv}_{\mathsf{DIBE},\mathcal{D}_k}(\mathcal{B})^{\mathsf{PR\text{-}ID\text{-}CPA}} \leq (\mathsf{Adv}_{\mathcal{D}_k,\mathsf{Setup}}(\mathcal{B}) + 2q_k(\mathsf{Adv}_{\mathcal{D}_k,\mathsf{Setup}}(\mathcal{B})) + 1/q).^3$$

[3] q_k is the maximal number of query to the Eval oracle

$\underline{\mathsf{Setup}(\mathfrak{K})}$:
$\mathbf{B} \xleftarrow{\$} \mathcal{D}_k$
For $i \in [\![1, \ell]\!] : y_i \xleftarrow{\$} \mathbb{Z}_p; z_i = y_i \cdot a \in \mathbb{Z}_p$
$y' \xleftarrow{\$} \mathbb{Z}_p; z' = y'^{\top} \cdot a \in \mathbb{Z}_p;$
$\mathsf{mpk} := (\mathcal{G}, [a]_1, ([z_i]_1)_{1 \leq i \leq \ell}, [z']_1)$
$\mathsf{msk} := ((y_i)_{1 \leq i \leq \ell}), [y']_1)$
Return $(\mathsf{mpk}, \mathsf{msk})$

$\underline{\mathsf{USKGen}(\mathsf{msk}, \mathsf{id})}$:
$t \xleftarrow{\$} \mathbb{Z}_p$
$v = y' + \sum_{i=1}^{\ell} \mathsf{id}_i y_i t$

$S \xleftarrow{\$} \mathbb{Z}_p; T = \mathbf{B} \cdot S \in \mathbb{Z}_p$
$V = \sum_{i=0}^{\ell(\mathsf{id})} \mathsf{id}_i \mathbf{Z}_i T$
For $i, \mathsf{id}[i] = 1$:
$e_i = y_i \cdot t \in \mathbb{Z}_p; E_i = \mathbf{Z}_i T \in \mathbb{Z}_p$

$\mathsf{usk} := ([t]_2, [v]_2)$
$\mathsf{udk}[\mathsf{id}] := ([T]_2, [V]_2, ([e_i]_2, [E_i]_2)_{i, \mathsf{id}[i]=1})$
Return $(\mathsf{usk}[\mathsf{id}], [e_i]_2)$

$\underline{\mathsf{Enc}(\mathsf{usk}[\mathsf{id}], \tilde{\mathsf{id}})}$:
$r \xleftarrow{\$} \mathbb{Z}_p^k; c_0 = \mathbf{A} r \in \mathbb{Z}_p^{k+1}$
$c_1 = (\sum_{i=0}^{\ell(\mathsf{id})} \mathsf{id} \mathbf{Z}_i) \cdot r \in \mathbb{Z}_p^n$
$K = z' \cdot r \in \mathbb{Z}_p$
Return $c = ([c_0]_1, [c_1]_1)$ et $\mathsf{sk} = [K]_T$.

$\underline{\mathsf{USKDown}(\mathsf{usk}[\mathsf{id}], \tilde{\mathsf{id}})}$:
If $\neg(\tilde{\mathsf{id}} \preceq \mathsf{id})$, return $\perp$.

Set $\mathcal{I} = \{i | \tilde{\mathsf{id}}[i] = 0 \wedge \mathsf{id}[i] = 1\}$
Downgrading the key :
$\hat{v} = v + \sum_{i \in \mathcal{I}} \mathsf{id}_{S_j, i} e_i \in \mathbb{Z}_p$
$\hat{V} = V + \sum_{i \in \mathcal{I}} \mathsf{id}_{S_j, i} E_i \in \mathbb{Z}_p$
Rerandomization of $(\hat{v}, \hat{V})$:
$s' \leftarrow \mathbb{Z}_p; S' \leftarrow \mathbb{Z}_p$
$t' = t + T s' \in \mathbb{Z}_p$
$T' = T \cdot S'$
$\hat{v}' = \hat{v} + \hat{V} \cdot s' \in \mathbb{Z}_p$
$V' = \hat{V} \cdot S' \in \mathbb{Z}_p$
Rerandomization of e_i:
For i, $\tilde{\mathsf{id}}[i] = 1$:
$e'_i = e_i + E_i s' \in \mathbb{Z}_p$
$E'_i = E_i \cdot S' \in \mathbb{Z}_p$

$\mathsf{usk}[\mathsf{id}'] = ([t']_2, [v']_2)$
$\mathsf{udk}[\mathsf{id}'] = ([T']_2, [V']_2, [e'_i]_2, [E'_i]_2)$

Return $(\mathsf{usk}[\mathsf{id}], \mathsf{udk}[\mathsf{id}'])$

$\underline{\mathsf{Verify}(\mathsf{udk}[\tilde{\mathsf{id}}], c)}$:
$\mathsf{usk}[\mathsf{id}] \stackrel{?}{=} ([t']_2, [v']_2)$ and $c \stackrel{?}{=} ([c_0]_1, [c_1]_1)$.
$\mathsf{sk} = e([c_0]_1, [v]_2) \cdot e([c_1]_1, [t]_2)^{-1}$
Return $\mathsf{sk} \in \mathbb{G}_T$

Fig. 9 DIBE from Blazy et al. (2019)