Designated Verifier Signatures: Anonymity and Efficient Construction from *any* Bilinear Map

Fabien Laguillaumie[1,2] and Damien Vergnaud[2]

[1] France Telecom Research and Development
42, rue des Coutures, B.P. 6243, 14066 Caen Cedex 4, France,
[2] Laboratoire de Mathématiques Nicolas Oresme
Université de Caen, Campus II, B.P. 5186,
14032 Caen Cedex, France,
{laguillaumie,vergnaud}@math.unicaen.fr

Abstract. The concept of *Designated Verifier Signatures (DVS)* was introduced by Jakobsson, Sako and Impagliazzo at Eurocrypt'96. These signatures are intended to a specific verifier, who is the only one able to check their validity. In this context, we formalize the notion of *privacy of signer's identity* which captures the *strong designated verifier property* investigated in their paper. We propose a variant of the pairing-based DVS scheme introduced at Asiacrypt'03 by Steinfeld, Bull, Wang and Pieprzyk. Contrary to their proposal, our new scheme can be used with any admissible bilinear map, especially with the low cost pairings and achieves the new anonymity property (in the random oracle model). Moreover, the unforgeability is *tightly* related to the Gap-Bilinear Diffie-Hellman assumption, in the random oracle model and the signature length is around 75 % smaller than the original proposal.

Keywords: Designated verifier signatures, Privacy of signer's identity, Bilinear Diffie-Hellman problems, Exact security, Tight reduction

1 Introduction.

Recently, Steinfeld, Bull, Wang and Pieprzyk [17] proposed a designated verifier signature scheme based on pairings. In this article, we propose three techniques which significantly improve this protocol. First of all, a novel use of a hash function in a context of digital signatures permits to rehabilitate the low cost pairing, namely the discrete exponentiation, which has been turned down because it suffers from some unavoidable drawbacks as a bilinear map. The efficiency is increased by a factor of 3.5 to 16, and the signature length is around 75 % smaller than the original proposal. Secondly, we formally define a notion of anonymity of signers, and, randomizing the signatures makes our scheme achieve this property. As a side effect, its unforgeability is *tightly* related to the Gap Bilinear Diffie-Hellman assumption. Finally, the proofs of security rely on a new use of a Decisional Bilinear Diffie-Hellman oracle in the simulation of a random oracle.

Related work The *self-authenticating* property of digital signatures can be suitable for some applications such as dissemination of official announcements, but it is sometimes undesirable in personally or commercially sensitive applications. Therefore it may be preferable to put some restrictions on this property to prevent potential misuses of signatures. To address this concern, several techniques that allow users to generate a signature with anonymity have been developed over the years. The concept of **D**esignated **V**erifier **S**ignatures (DVS) was introduced by Jakobsson, Sako and Impagliazzo at Eurocrypt'96 [10] and independently by Chaum in the patent [7], under the name of *private signatures*. They are intended to a specific and unique designated verifier, who is the only one able to check their validity. This verifier cannot convince another party that the signature is actually valid, essentially because he can also perform this signature by himself. This means, in particular, that DVS do not provide the main property of ordinary digital signatures, namely non-repudiation. As explained in [10], in some cases, it may be desirable that DVS provide an even stronger notion of privacy: given a DVS and two potential signing public keys, it is computationally infeasible for an eavesdropper to determine under which of the two corresponding secret keys the signature was performed. Following [10], we call *strong designated verifier signatures*, the DVS schemes that achieve this property.

In [14], Rivest, Shamir and Tauman introduced the ring signatures (see also [6]). By setting the size of the ring to two members, these signatures provide DVS. Many ring signatures have been proposed but they do not achieve the strong designated verifier property. Recently, in [15], Saeednia, Kremer and Markowitch proposed very efficient DVS with signatures *à la* Schnorr. They proved the existential unforgeability of their scheme under a no-message attack and argued that their scheme performs the strong designated verifier property (this property is defined in terms of simultability). But lacking a good security model, they could not prove that their scheme achieves these security notions under adaptive chosen-message attack. In [19], Susilo, Zhang and Mu proposed an identity-based strong DVS which is a pairing-based variant of [15] and whose security is investigated in the same model. In [17], Steinfeld, Bull, Wang and Pieprzyk proposed a formalization of *Universal DVS* (UDVS). These are ordinary digital signatures with the additional functionality that any holder of a signature is able to convert it into a DVS specified to any designated verifier of his choice. Moreover they showed that bilinear maps allow an elegant construction of a UDVS scheme (DVSBM). A similar construction has been proposed independently by the authors in [11]. At PKC'04 [18], Steinfeld, Wang and Pieprzyk proposed a slightly stronger security model, which allows the attacker, while mounting a chosen-message attack, to query the verification of any couple message/signature of its choice. In their article they give three new DVS constructions based on Schnorr and RSA signatures.

Our contributions In this paper, we formalize the notion of *privacy of signer's identity* which captures the strong designated verifier property. For public-key encryption, Bellare, Boldyreva, Desai and Pointcheval defined, in [1], an additional security requirement which includes the notion that an attacker cannot determine under which key an encryption was performed: it is the idea of *key-privacy*. Our formalization follows this notion. Steinfeld *et al.* proposed at Asiacrypt'03 [17] an interesting and promising scheme based on pairing, which however suffers from a lack of efficiency (compared to [15]'s scheme for instance). Moreover their scheme is not secure with low cost pairings.

We revise it such that, at equal security guarantees, we obtain the most efficient UDVS scheme, and instantiated with the discrete exponentiation we obtain the most efficient DVS protocol in practice (*cf.* Section 4.2), but loose the universal property. The first modification which consists in a novel use of hash function in the asymmetric signature setting makes it possible to shorten the signatures and allows the scheme to be used with any admissible bilinear map. Short signatures are useful for low-bandwidth devices and environments where a person is asked to manually type in the signature. By using this technique, for a security level of 2^{80} bit operations, the signature length is 271 bits and does not depend on the size of the ground field. The second trick consists in making the signature generation not deterministic. With this randomization we can draw a scheme which achieves privacy of signer's identity under an adaptive chosen-message attack in the random oracle model [3]. As in [8], it also makes the unforgeability of the modified scheme *tightly* related to the underlying problem, in the random oracle model. We introduce a new use of a Decisional Bilinear Diffie-Hellman oracle in the security proofs to maintain a random oracle list. We obtain a very tight link between the security of the scheme and the Gap Bilinear Diffie-Hellman assumption, with a quadratic time reduction.

In the rest of the paper, we recall the definition of DVS, then we formalize the new anonymity requirement for DVS. In section 4, we present our new scheme with a precise security treatment. In appendix, we discuss the security of some other schemes.

2 Definition and security assumptions for designated verifier signatures

In this section, we state the definition of DVS schemes induced by Steinfeld *et al.*'s formalization.

Definition 1 (Designated Verifier Signature Scheme). *Given an integer k, a* (weak) designated verifier signature scheme *`DVS` with security parameter k is defined by the following:*

- *a* common parameter generation algorithm *`DVS.Setup`: it is a probabilistic algorithm which takes k as input. The outputs are* the public parameters*;*
- *a* signer key generation algorithm *`DVS.SKeyGen`: it is a probabilistic algorithm which takes the public parameters as input and outputs a pair of signing keys $(\mathbf{pk_A}, \mathbf{sk_A})$;*
- *a* verifier key generation algorithm *`DVS.VKeyGen`: it is a probabilistic algorithm which takes the public parameters as inputs, and outputs a pair of verifying keys $(\mathbf{pk_B}, \mathbf{sk_B})$;*

– *a* designated verifier signing algorithm ***DVS.Sign****: it takes a message m, a signing secret key $\mathbf{sk_A}$, a verifying public key $\mathbf{pk_B}$ and the public parameters as inputs . The output σ is* a B-designated verifier signature of m. *This algorithm can be either probabilistic or deterministic;*
– *a* designated verifying algorithm ***DVS.Verify****: it is a deterministic algorithm which takes a bit string σ, a message m, a signing public key $\mathbf{pk_A}$, a verifying secret key $\mathbf{sk_B}$ and the public parameters as inputs, and tests whether σ is a valid B-designated verifier signature of m with respect to the keys* $(\mathbf{pk_A}, \mathbf{sk_A}, \mathbf{pk_B}, \mathbf{sk_B})$.

Moreover, a designated verifier signature scheme must satisfy the following properties (formally defined in [18] and discussed below):

1. correctness*: a properly formed B-designated verifier signature must be accepted by the verifying algorithm;*
2. unforgeability*: given a pair of signing keys* $(\mathbf{pk_A}, \mathbf{sk_A})$ *and a pair of verifying keys* $(\mathbf{pk_B}, \mathbf{sk_B})$*, it is computationally infeasible, without the knowledge of the secret key $\mathbf{sk_A}$ or $\mathbf{sk_B}$, to produce a valid B-designated verifier signature;*
3. source hiding*: given a message m and a B-designated verifier signature σ of this message, it is (unconditionally) infeasible to determine who from the original signer or the designated verifier performed this signature, even if one knows all secrets;*

For digital signatures, the widely accepted notion of security was defined by Goldwasser, Micali and Rivest in [9] as *existential forgery against adaptive chosen-message attack* (EF-CMA). For a DVS scheme, the security model proposed in [17] and [18] (under the designation ST-DV-UF) is similar, with the notable difference that, while mounting a chosen-message attack, we allow the attacker to query a verifying oracle on any couple message/signature of its choice. As usual, in the adversary answer, there is the natural restriction that the returned message/signature has not been obtained from the signing oracle (for more details, we refer the reader to [17] and [18]). In order to be consistent with the classical security model for usual signatures, also for DVS we denote this security point by EF-CMA.

In their formalization of UDVS [17] [18], Steinfeld *et al.* defined the *Non-Transferability Privacy* property to prevent a designated-verifier from using a DVS to produce evidence which convinces a third-party that this DVS was actually computed by the signer. However, their notion is computational, and we believe that the identity of the signer should be unconditionally protected (*i.e.* DVS should provide information theoretical anonymity), as in ring signatures (where this security requirement is called *source hiding*).

Finally, even with this unconditional ambiguity, anyone can check that there are only two potential signers for a DVS. If signatures are captured on the line before reaching the verifier, an eavesdropper will be convinced that the designated verifier did not produce the signature. Therefore, in [10], Jakobsson *et al.* suggested a stronger notion of anonymity:

Definition 2 (Strong Designated Verifier Signature Scheme). *Given an integer k, a* strong designated verifier signature scheme *$\mathtt{DVS}$ with security parameter k is a designated verifier signature scheme with security parameter k, which satisfies the following additional property (formally defined in the next section):*

4. *privacy of signer's identity: given a message m and a B-designated verifier signature σ of this message, it is computationally infeasible, without the knowledge of the secret key of B or the one of the signer, to determine which pair of signing keys was used to generate σ.*

3 Anonymity of DVS

3.1 Formal definition

In this section, we define formally the *privacy of signer's identity* under a chosen message attack (PSI-CMA). We consider a PSI-CMA-adversary $\mathcal{A}$ in the random oracle model, which runs in two stages: in the `find` stage, it takes two signing public keys $\mathbf{pk_{A_0}}$ and $\mathbf{pk_{A_1}}$ and a verifying public key $\mathbf{pk_B}$, and outputs a message $m^\star$ together with some state information $\mathcal{I}^\star$. In the `guess` stage, it gets a challenge B-designated verifier signature $\sigma^\star$ formed by signing the message $m^\star$ at random under one of the two keys and the information $\mathcal{I}^\star$, and must say which key was chosen. The adversary has access to the random oracle(s) $\mathcal{H}$, to the signing oracles $\Sigma_{A_0,B}$, $\Sigma_{A_1,B}$ and to the verifying oracle Υ_B, and is allowed to invoke them on any message with the restriction of not querying $(m^\star, \sigma^\star)$ from the verifying oracle in the second stage.

Definition 3 (Privacy of signer's identity). *Let k be an integer and $\mathtt{DVS}$ a designated verifier signature scheme with security parameter k. We consider the following random experiment, for $r \in \{0,1\}$:*

$\boxed{\textit{Experiment } \mathbf{Exp}^{\textsf{psi-cma}-r}_{\mathtt{DVS},\mathcal{A}}(k)}$
$\text{params} \xleftarrow{R} \mathtt{DVS.Setup}(k)$
$(\mathbf{pk_{A_0}}, \mathbf{sk_{A_0}}) \xleftarrow{R} \mathtt{DVS.SKeyGen}(\text{params})$
$(\mathbf{pk_{A_1}}, \mathbf{sk_{A_1}}) \xleftarrow{R} \mathtt{DVS.SKeyGen}(\text{params})$
$(\mathbf{pk_B}, \mathbf{sk_B}) \xleftarrow{R} \mathtt{DVS.VKeyGen}(\text{params})$
$(m^\star, \mathcal{I}^\star) \leftarrow \mathcal{A}^{\Sigma_{A_0,B},\Sigma_{A_1,B},\Upsilon_B,\mathcal{H}}(\mathtt{find}, \mathbf{pk_B}, \mathbf{pk_{A_0}}, \mathbf{pk_{A_1}})$
$\sigma^\star \leftarrow \mathtt{DVS.Sign}(\text{params}, m^\star, \mathbf{sk_{A_r}}, \mathbf{pk_B})$
$d \leftarrow \mathcal{A}^{\Sigma_{A_0,B},\Sigma_{A_1,B},\Upsilon_B,\mathcal{H}}(\mathtt{guess}, m^\star, \mathcal{I}^\star, \sigma^\star, \mathbf{pk_B}, \mathbf{pk_{A_0}}, \mathbf{pk_{A_1}})$
Return d

We define the advantage *of the adversary $\mathcal{A}$,* via

$$\mathbf{Adv}^{\mathsf{psi-cma}}_{\mathtt{DVS},\mathcal{A}}(k) = \left| Pr\left[\mathbf{Exp}^{\mathsf{psi-cma}-1}_{\mathtt{DVS},\mathcal{A}}(k) = 1\right] - Pr\left[\mathbf{Exp}^{\mathsf{psi-cma}-0}_{\mathtt{DVS},\mathcal{A}}(k) = 1\right]\right|.$$

Let $t \in \mathbb{N}$ and $\varepsilon \in [0,1]$. The scheme $\mathtt{DVS}$ is said to be (k,t,ε)-PSI-CMA secure, if the function $\mathbf{Adv}^{\mathsf{psi-cma}}_{\mathtt{DVS},\mathcal{A}}(k)$ is smaller than ε for any PSI-CMA-adversary $\mathcal{A}$ running in time complexity less than t.

3.2 Semantically secure encryption implies anonymity

In [10], Jakobsson *et al.* suggested that "in order to make protocols strong designated verifier, transcripts can be probabilistically encrypted using the public key of the intended verifier". This is not sufficient in general (for instance a plaintext El Gamal encryption does not protect the anonymity of the signers). However, in this paragraph, we prove that using an additional IND-CCA2 public-key encryption layer is actually sufficient to make any DVS scheme strong.

Basically, being able to distinguish two potential signing keys in the signature scheme will give an advantage to distinguish two potential encrypted messages.

Let k be an integer, let `DVS` be a (weak)-designated verifier signature scheme with security parameter k and let Π be any IND-CCA2 encryption scheme. We define a designated verifier signature $\mathtt{DVS}^{\Pi}$ as follows: the generation of a $\mathtt{DVS}^{\Pi}$ signature of a message m is done by encrypting a `DVS` signature σ of m under the designated verifier public key. Its verification is performed by first decrypting the signature, then verifying it with the `DVS.Verify` algorithm.

Proposition 1. *Let k be an integer, let DVS be a (weak)-designated verifier signature scheme with security parameter k, and let Π be an IND-CCA2 encryption scheme with security parameter k. Then DVS^{Π} is a strong designated verifier signature scheme. More precisely, for any PSI-CMA adversary $\mathcal{A}$ with security parameter k which takes advantage $\mathbf{Adv}^{\mathsf{psi-cma}}_{DVS^{\Pi},\mathcal{A}}$ against DVS^{Π} within time t, making $q_{\mathcal{H}}$, q_{Σ} and q_{Υ} queries to the random oracle(s), the signing oracle and the verifying oracle respectively, there exists an IND-CCA2 adversary $\mathcal{A}'$ against Π, making $q_{\mathcal{H}}$ queries to the random oracle(s), and q_{Υ} queries to the decrypting oracle, within time t, which has the same advantage as $\mathcal{A}$.*

Proof (sketch). A general study of the security notions and attacks for encryption schemes was conducted in [2]. We refer the reader to this paper for the definition of IND-CCA2 encryption.

We construct the algorithm $\mathcal{A}'$ as follows:

- $\mathcal{A}'$ is fed with a public key $\mathbf{Epk_B}$ for Π and chooses two pairs of signing keys $(\mathbf{sk_{A_0}}, \mathbf{pk_{A_0}})$ $(\mathbf{sk_{A_1}}, \mathbf{pk_{A_1}})$ and a pair of verifying keys $(\mathbf{sk_B}, \mathbf{pk_B})$.
- $\mathcal{A}$ is fed with $\mathbf{Epk_B}$, $\mathbf{pk_B}$, $\mathbf{pk_{A_0}}$ and $\mathbf{pk_{A_1}}$.
- In both stages, for any signing query from $\mathcal{A}$, $\mathcal{A}'$ answers using the secret key of either A_0 or A_1. For any verifying query from $\mathcal{A}$, $\mathcal{A}'$ answers using the secret key $\mathbf{Dpk_B}$ of B and the decryption oracle.
- Eventually, in the `find` stage, $\mathcal{A}$ outputs a message $m \in \{0,1\}^*$.
- $\mathcal{A}'$ computes two pre-signatures σ_0 and σ_1 using the `DVS.Sign` algorithm of the message m, and queries these signatures to the IND-CCA2 challenger which answers with an encryption of σ_b where $b \in_R \{0,1\}$.
- $\mathcal{A}'$ gives this challenge to $\mathcal{A}$ as the answer to the PSI-CMA challenge. The only verification query that $\mathcal{A}'$ cannot answer is the one $\mathcal{A}$ is not allowed to ask.
- Finally $\mathcal{A}$ outputs a bit b' in the `guess` stage.

By definition of $\mathcal{A}$, $b' = b$ with probability $\mathbf{Adv}^{\mathsf{psi-cma}}_{\mathsf{DVS}^{\Pi},\mathcal{A}}$ and $\mathcal{A}'$ distinguishes the two messages σ_0 and σ_1 encrypted by Π with the same advantage $\mathbf{Adv}^{\mathsf{ind-cca}}_{\Pi,\mathcal{A}'} = \mathbf{Adv}^{\mathsf{psi-cma}}_{\mathsf{DVS}^{\Pi},\mathcal{A}}$. This concludes the proof.

4 The new scheme : DVSBMH

4.1 Bilinear maps and underlying problems.

In this section, we recall some definitions concerning bilinear maps.

Definition 4 (Admissible bilinear map [4]). *Let $(\mathbb{G}_0, +)$, $(\mathbb{G}_1, +)$ and $(\mathbb{H}, \times)$ be three groups of the same prime order q and let P_0 and P_1 be two generators of $\mathbb{G}_0$ and $\mathbb{G}_1$ (respectively). An* admissible bilinear map *is a map $e : \mathbb{G}_0 \times \mathbb{G}_1 \longrightarrow \mathbb{H}$ satisfying the following properties:*

- *bilinear: $e(aQ, bR) = e(Q, R)^{ab}$ for all $(Q, R) \in \mathbb{G}_0 \times \mathbb{G}_1$ and all $(a, b) \in \mathbb{Z}^2$;*
- *non-degenerate: $e(P_0, P_1) \neq 1$;*
- *computable: there exists an efficient algorithm to compute e.*

Definition 5 (prime-order-BDH-parameter-generator [4]). *A* prime-order-BDH-parameter-generator *is a probabilistic algorithm that takes a security parameter k as input and outputs a 7-tuple $(q, P_0, \mathbb{G}_0, P_1, \mathbb{G}_1, \mathbb{H}, e)$ satisfying the following conditions: q is a prime with $2^k < q < 2^{k+1}$, the groups $\mathbb{G}_0, \mathbb{G}_1$ and $\mathbb{H}$ are of order q, P_0 and P_1 generates $\mathbb{G}_0$ and $\mathbb{G}_1$ (respectively), and $e : \mathbb{G}_0 \times \mathbb{G}_1 \longrightarrow \mathbb{H}$ is an admissible bilinear map. A prime-order-BDH-parameter-generator $\mathcal{G}en$ is said to be* symmetric *if $P_0 = P_1$ and $\mathbb{G}_0 = \mathbb{G}_1$ for any 7-tuple $(q, P_0, \mathbb{G}_0, P_1, \mathbb{G}_1, \mathbb{H}, e)$ output by $\mathcal{G}en$.*

Let $(\mathbb{G}_0, +)$, $(\mathbb{G}_1, +)$ and $(\mathbb{H}, \times)$ be three groups of the same large prime order q, P_0 and P_1 be two generators of $\mathbb{G}_0$ and $\mathbb{G}_1$ (respectively), and let $e : \mathbb{G}_0 \times \mathbb{G}_1 \longrightarrow \mathbb{H}$ be an admissible bilinear map. For most of the applications of pairings in cryptography, it is necessary to know an efficient way to compute an isomorphism $\varphi : \mathbb{G}_0 \simeq \mathbb{G}_1$. Contrary to Weil or Tate pairings, this is not true for the discrete exponentiation $e : \langle P_0 \rangle \times (\mathbb{Z}/q\mathbb{Z}, +) \longrightarrow \langle P_0 \rangle, (P, x) \longmapsto xP$ where the map $\langle P_0 \rangle \longrightarrow \mathbb{Z}/q\mathbb{Z}$ is the discrete logarithm.

At PKC'01, Okamoto and Pointcheval proposed a new class of computational problems, called gap problems [13]. Essentially, a gap problem is a dual to inverting and decisional problems. More precisely, this problem is to solve an inverting problem with the help of an oracle for a decisional problem. Following this idea, we state the following problems (where $\mathbb{G}_0$ and $\mathbb{G}_1$ have not a symmetric role):

Computational Bilinear Diffie-Hellman (CBDH): let a, b and c be three integers. Given aP_0, bP_0, cP_1, compute $e(P_0, P_1)^{abc}$.
Decisional Bilinear Diffie-Hellman (DBDH): let a, b, c and d be four integers. Given aP_0, bP_0, cP_1 and $e(P_0, P_1)^d$, decide whether $d = abc \mod q$.
Gap-Bilinear Diffie-Hellman (GBDH): let a, b and c be three integers. Given aP_0, bP_0, cP_1, compute $e(P_0, P_1)^{abc}$ with the help of a DBDH Oracle.

Definition 6 (CBDH and GBDH assumption). *Let $\mathcal{G}en$ be a prime-order-BDH-parameter-generator. Let D be an adversary that takes as input a 7-tuple $(q, P_0, \mathbb{G}_0, P_1, \mathbb{G}_1, \mathbb{H}, e)$ generated by $\mathcal{G}en$ and $(X, Y, Z) \in \mathbb{G}_0^2 \times \mathbb{G}_1$. He returns an element of $h \in \mathbb{H}$. We consider the following random experiments, where k is a security parameter and $\mathcal{O}_{DBDH}$ is a DBDH oracle:*

$\boxed{\textit{Experiment } \mathbf{Exp}^{\mathsf{cbdh}}_{\mathcal{G}en,D}(k)}$	$\boxed{\textit{Experiment } \mathbf{Exp}^{\mathsf{gbdh}}_{\mathcal{G}en,D}(k)}$
$(q, P_0, \mathbb{G}_0, P_1, \mathbb{G}_1, \mathbb{H}, e) \xleftarrow{R} \mathcal{G}en(k)$	$(q, P_0, \mathbb{G}_0, P_1, \mathbb{G}_1, \mathbb{H}, e) \xleftarrow{R} \mathcal{G}en(k)$
$\texttt{setup} \leftarrow (q, P_0, \mathbb{G}_0, P_1, \mathbb{G}_1, \mathbb{H}, e)$	$\texttt{setup} \leftarrow (q, P_0, \mathbb{G}_0, P_1, \mathbb{G}_1, \mathbb{H}, e)$
$x \xleftarrow{R} [\![1, q-1]\!]$, $X \leftarrow xP_0$	$x \xleftarrow{R} [\![1, q-1]\!]$, $X \leftarrow xP_0$
$y \xleftarrow{R} [\![1, q-1]\!]$, $Y \leftarrow yP_0$	$y \xleftarrow{R} [\![1, q-1]\!]$, $Y \leftarrow yP_0$
$z \xleftarrow{R} [\![1, q-1]\!]$, $Z \leftarrow zP_1$	$z \xleftarrow{R} [\![1, q-1]\!]$, $Z \leftarrow zP_1$
$h \leftarrow D(\texttt{setup}, X, Y, Z)$	$h \leftarrow D^{\mathcal{O}_{DBDH}}(\texttt{setup}, X, Y, Z)$
Return 1 *if* $h = e(P_0, P_1)^{xyz}$,	*Return* 1 *if* $h = e(P_0, P_1)^{xyz}$,
0 *otherwise*	0 *otherwise*

We define the success *of D in solving the CBDH and the GBDH problems* via

$$\mathbf{Succ}^{\mathsf{cbdh}}_{\mathcal{G}en,D}(k) = \Pr[\mathbf{Exp}^{\mathsf{cbdh}}_{\mathcal{G}en,D}(k) = 1] \textit{ and } \mathbf{Succ}^{\mathsf{gbdh}}_{\mathcal{G}en,D}(k) = \Pr[\mathbf{Exp}^{\mathsf{gbdh}}_{\mathcal{G}en,D}(k) = 1]$$

Let t be an integer and ε a real in $[0, 1]$. $\mathcal{G}en$ is said to be (k, t, ε)-CBDH-secure (resp. (k, t, ε)-GBDH-secure) if no adversary D running in time t has success $\mathbf{Succ}^{\mathsf{cbdh}}_{\mathcal{G}en,D}(k) \geq \varepsilon$ (resp. $\mathbf{Succ}^{\mathsf{gbdh}}_{\mathcal{G}en,D}(k) \geq \varepsilon$).

Notations : we denote by $T_{\mathrm{Exp}-\mathbb{G}}$ the time complexity for evaluating exponentiation in a group $\mathbb{G}$ and $T_{\mathcal{O}}$ the time complexity of the oracle $\mathcal{O}$.

4.2 Description of the new scheme DVSBMH

The scheme DVSBM, proposed at Asiacrypt'03 by Steinfeld *et al.* [17] is a pairing-based DVS. The signature generation is deterministic, therefore this scheme can certainly not achieve the PSI-CMA security point. The authors required that the isomorphism between $\mathbb{G}_0$ and $\mathbb{G}_1$ is known and efficiently computable. In fact, DVSBM is trivially *not* secure if we use the discrete exponentiation.

We introduce a variant of DVSBM which is more efficient, achieves the property of privacy of signer's identity and whose security is proven even if we use the discrete exponentiation. For industrial purposes, where efficiency prevails over exact security, the choice of the parameters is oriented by the underlying algorithmic problems without consideration of the reduction cost in the security proof (we call it *industrial security*). Considering the best algorithms to solve GBDH in both settings, the scheme with the discrete exponentiation will be prefered in practice, whereas the scheme with the Weil or Tate pairing has a tighter security reduction.

In DVSBM, the verification of signatures consists only in checking an equality between two quantities which can be computed independently by the signer and the verifier, it is actually sufficient to check the equality of some hash values of these quantities. This first remark, which seems to have been overlooked in [17], makes it possible to shorten the signature considerably and to use the discrete exponentiation to instantiate the protocol.

Our second trick aims at randomizing the signature. We prove that this is sufficient to obtain the anonymity of signers. Moreover, the security of the signature is *tightly* related to the GBDH and this random salt ensures the anonymity of signers. Using these tricks, we define DVSBMH.

Description of DVSBMH

Setup Let k be a security parameter. Let $\mathcal{G}en$ be a prime-order-BDH-parameter-generator, f_1, f_2, $f_r : \mathbb{N} \to \mathbb{N}$ be three functions. We denote $k_1 = f_1(k)$, $k_2 = f_2(k)$ and $n_r = f_r(k)$. Let $(q, P_0, \mathbb{G}_0, P_1, \mathbb{G}_1, \mathbb{H}, e)$ be a 7-tuple generated by $\mathcal{G}en(k_1)$. Let $[\{0,1\}^* \times \{0,1\}^{n_r} \longrightarrow \mathbb{G}_1]$ be a hash function family, and h be a random member of this family. Let $[\mathbb{H} \longrightarrow \{0,1\}^{k_2}]$ be a hash function family, and g be a random member of this family.

SKeyGen $a \in [\![1, q-1]\!]$ is the secret key, $P_A = aP_0$ is the public one
VKeyGen $b \in [\![1, q-1]\!]$ is the secret key, $P_B = bP_0$ is the public one

Sign Given a message m, the secret key a of the signer, the public key P_B of the designated verifier, compute $H = h(m, r)$ for some random string r of length n_r and $s = g(e(P_B, aH))$ and the signature is $\sigma = (r, s)$.

Verify Given a pair $(m, (s, r))$, the signer's public key P_A, and the verifier's secret key b, the algorithm accepts the signature if and only if $s = g(e(P_A, bh(m, r)))$.

In practice, for a security requirement of 2^{80} operations (*i.e.* $k = 80$), we use the values $k_1 = k_2 = 160$ and $n_r = 111$ which are derived from the security proofs (*cf.* [12]). The correctness and source hiding properties of DVSBMH are straightforward. In general, the new scheme does not satisfy the universal property from [17] any more, because the security of BLS signatures [5] relies on the existence of an efficiently computable isomorphism from $\mathbb{G}_0$ to $\mathbb{G}_1$.

4.3 Security of DVSBMH when $\mathbb{G}_0 = \mathbb{G}_1$

Here we formally investigate the security of the version of DVSBMH for which we know an algorithm to compute the isomorphism between $\mathbb{G}_0$ and $\mathbb{G}_1$ in the random oracle model (*i.e.* we replace the hash functions h and g by random oracles $\mathcal{H}$ and $\mathcal{G}$). For simplicity, we assume $\mathbb{G}_0 = \mathbb{G}_1 = \mathbb{G}$. In practice such a setting can be obtained with, for instance, the Weil or Tate pairing. In this case our new scheme can be extended to a UDVS scheme related to the randomized BLS signatures [5, 8]. This is an important consideration because we prove that the unforgeability is *tightly* related to the GBDH problem, therefore this scheme offers the best exact security of all DVS protocols. Moreover, it achieves the privacy of signer's identity under the CBDH assumption (with the random salt but without the g hash function, the anonymity would have been related to DBDH, an easier problem). These results are described in the following theorems.

Theorem 1 (Unforgeability of DVSBMH). *Let $\mathcal{G}en$ be a symmetric prime-order-BDH-parameter-generator, let f_1, f_2, $f_r : \mathbb{N} \to \mathbb{N}$ be three functions and let DVSBMH be the associated DVS scheme. For any EF-CMA-adversary $\mathcal{A}$, in the random oracle model, against DVSBMH, with security parameter k which has success $\varepsilon = \mathbf{Succ}^{\mathsf{ef-cma}}_{\mathit{DVSBMH},\mathcal{A}}(k)$, running time t, and makes $q_{\mathcal{H}}$ and $q_{\mathcal{G}}$ queries to the random oracles, q_{Σ} queries to the signing oracles and q_{Υ} queries to the verifying oracle, there exists an adversary D for GBDH which has advantage $\varepsilon' = \mathbf{Succ}^{\mathsf{gbdh}}_{\mathcal{G}en,D}(k)$ running in time $t' \in \mathbb{N}$ such that*

$$\begin{cases} \varepsilon' \ \geq \ \varepsilon - \dfrac{(q_{\mathcal{H}} + q_{\Sigma})q_{\Sigma}}{2^{n_r}} - (1 + q_{\Upsilon})\left(\dfrac{q_{\mathcal{G}}}{2^{k_1}} + \dfrac{1}{2^{k_2}}\right) \\ t' \leq t + (q_{\mathcal{H}} + q_{\Sigma})\left(T_{Exp-\mathbb{G}} + O(1)\right) + q_{\Sigma}\left(T_{Exp-\mathbb{H}} + O(1)\right) \\ \qquad\quad + (q_{\Upsilon} + 1)\left(T_{DBDH} + O(1)\right) \end{cases}$$

where $k_1 = f_1(k)$, $k_2 = f_2(k)$ and $n_r = f_r(k)$.

Proof. The proof is a straightforward modification of the proof of [17] using the additional technique in [8]. Due to the lack of space, we have not written it down.

Theorem 2 (Anonymity of DVSBMH). *Let $\mathcal{G}en$ be a symmetric prime-order-BDH-parameter-generator, let f_1, f_2, $f_r : \mathbb{N} \to \mathbb{N}$ be three functions and let DVSBMH be the associated DVS scheme. For any PSI-CMA-adversary $\mathcal{A}$, in the random oracle model, against DVSBMH, with security parameter k which has advantage $\varepsilon = \mathbf{Adv}^{\mathsf{psi-cma}}_{\mathit{DVSBMH},\mathcal{A}}(k)$, running time t, and makes $q_{\mathcal{H}}$ and $q_{\mathcal{G}}$ queries to the random oracles, q_{Σ} queries to the signing oracles and q_{Υ} queries to the verifying oracle, there exists an adversary D for CBDH which has advantage $\varepsilon' = \mathbf{Succ}^{\mathsf{cbdh}}_{\mathcal{G}en,D}(k)$ running in time $t' \in \mathbb{N}$ such that*

$$\begin{cases} \varepsilon' \geq \dfrac{\varepsilon}{2q_{\mathcal{G}}} - \dfrac{(q_{\mathcal{H}} + q_{\Sigma} + 1)(q_{\Sigma} + 1)}{2^{n_r} q_{\mathcal{G}}} - \dfrac{q_{\Upsilon}}{2^{k_2} q_{\mathcal{G}}} - \dfrac{q_{\mathcal{G}} q_{\Upsilon}}{2^{k_1} q_{\mathcal{G}}} \\ t' \leq t + (q_{\mathcal{H}} + q_{\Sigma})(T_{Exp-\mathbb{G}} + O(1)) + (q_{\Sigma} + q_{\Upsilon})(T_{Exp-\mathbb{H}} + O(1)) \end{cases}$$

where $k_1 = f_1(k)$, $k_2 = f_2(k)$ and $n_r = f_r(k)$.

Proof. Due to lack of space, the proof will be given in the full version of the paper [12].

4.4 Security of the general scheme

It is not necessary, thanks to our construction, to know explicitely an isomorphism between $\mathbb{G}_0$ and $\mathbb{G}_1$ to achieve a secure scheme. In this general case, we have a leak in terms of exact security compared to the previous case. In fact, we obtain a very tight link between the success probability of the adversary and the success in solving the GBDH problem but the reduction is quadratic time. When we use the discrete exponentiation as the underlying pairing (and so without the isomorphism), we get the best *industrial security*. We provide here the proof of the unforgeability, with the use of a decisional oracle to maintain the random oracle lists. The proof of the anonymity follows the same lines.

Theorem 3 (Unforgeability of DVSBMH). *Let $\mathcal{G}en$ be a prime-order-BDH-parameter-generator, let f_1, f_2, $f_r : \mathbb{N} \to \mathbb{N}$ be three functions and let DVSBMH be the associated DVS scheme. For any EF-CMA-adversary $\mathcal{A}$, in the random oracle model, against DVSBMH, with security parameter k which has success $\varepsilon = \mathbf{Succ}^{\mathsf{ef-cma}}_{DVSBMH,\mathcal{A}}$, running time t, and makes $q_{\mathcal{H}}$ and $q_{\mathcal{G}}$ queries to the random oracles, q_{Σ} queries to the signing oracles and q_{Υ} queries to the verifying oracle, there exists an adversary D for GBDH which has success $\varepsilon' = \mathbf{Succ}^{\mathsf{gbdh}}_{\mathcal{G}en,D}(k)$ running in time $t' \in \mathbb{N}$ such that*

$$\begin{cases} \varepsilon' \ \geq \ \varepsilon - \dfrac{q_{\Sigma}(q_{\mathcal{H}} + q_{\Sigma} + 1)}{2^{n_r}} - \dfrac{(q_{\mathcal{G}} + q_{\Sigma} + 1)(q_{\Sigma} + q_{\Upsilon} + 1)}{2^{k_1}} - \dfrac{(q_{\Upsilon} + 1)(q_{\Sigma} + 1)}{2^{k_2}} \\ t' \leq t + (q_{\mathcal{H}} + 2q_{\Sigma} + 1)\,(T_{Exp-\mathbb{G}_1} + O(1)) \\ \qquad\quad + (q_{\mathcal{G}} + q_{\Sigma} + 1)(q_{\mathcal{G}} + q_{\Sigma} + q_{\Upsilon})\,(T_{DDH} + O(1))\,, \end{cases}$$

where $k_1 = f_1(k)$, $k_2 = f_2(k)$ and $n_r = f_r(k)$.

Proof. The method of our proof is inspired by Shoup [16]: we define a sequence of games of modified attacks starting from the actual adversary. Let k be a security parameter, let $(q, P_0, \mathbb{G}_0, P_1, \mathbb{G}_1, \mathbb{H}, e)$ be a 7-tuple generated by $\mathcal{G}en(k_1)$ and (R_1, R_2, R_3) be a random instance of the GBDH problem.

Game_0 We consider an EF-CMA-adversary $\mathcal{A}$ with success $\varepsilon = \mathbf{Succ}^{\mathsf{ef-cma}}_{\mathsf{DVSBMH},\mathcal{A}}(k)$, within time t. The key generation algorithms are run and produce two pairs of keys $(\mathbf{sk_A}, \mathbf{pk_A})$ and $(\mathbf{sk_B}, \mathbf{pk_B})$. The adversary $\mathcal{A}$ is fed with $\mathbf{pk_A}$ and $\mathbf{pk_B}$ and, querying the random oracles $\mathcal{H}$ and $\mathcal{G}$, the signing oracle $\Sigma_{A,B}$ and the verifying oracle $\Upsilon_{A,B}$, it outputs a $(m^\star, (r^\star, s^\star))$ pair.
We denote by $q_{\mathcal{H}}$, $q_{\mathcal{G}}$, q_{Σ} and q_{Υ} the numbers of queries from the random oracles $\mathcal{H}$ and $\mathcal{G}$, from the signing oracle $\Sigma_{A,B}$ and from the verifying oracle $\Upsilon_{A,B}$. The only requirement is that the output signature $(r^\star, s^\star)$ has not been obtained from the signing oracle. When the adversary outputs its forgery, it can be checked whether it is actually valid or not. In any Game_j, we denote by $\mathsf{Forge_j}$ the event $\mathsf{DVSBMH}.\mathtt{Verify}(m^\star, (r^\star, s^\star), \mathbf{sk_B}, \mathbf{pk_A}) = 1$.
By definition, we have $\Pr[\mathsf{Forge_0}] = \mathbf{Succ}^{\mathsf{ef-cma}}_{\mathsf{DVSBMH},\mathcal{A}}(k)$.

Game_1 We modify the simulation by replacing $\mathbf{pk_A}$ by R_1 and $\mathbf{pk_B}$ by R_2. The distribution of $(\mathbf{pk_A}, \mathbf{pk_B})$ is unchanged since (R_1, R_2, R_3) is a random instance of the GBDH problem. Therefore we have $\Pr[\mathsf{Forge_1}] = \Pr[\mathsf{Forge_0}]$.

Game_2 In this game, we simulate the random oracle $\mathcal{H}$. For any fresh query $(m, r) \in \{0,1\}^* \times \{0,1\}^{n_r}$ to the oracle $\mathcal{H}$, we pick $u \in [\![1, q-1]\!]$ at random and compute $Q = uR_3$. We store (m, r, u, Q) in the $\mathsf{H\text{-}List}$ and return Q as the answer to the oracle call. In the random oracle model, this game is clearly identical to the previous one. Hence, $\Pr[\mathsf{Forge}_2] = \Pr[\mathsf{Forge}_1]$.

Game_3 We simulate the random oracle $\mathcal{G}$ by maintaining an appropriate $\mathsf{G\text{-}List}$. For any query $\tilde{s} \in \mathbb{H}$,

- we check whether the $\mathsf{G\text{-}List}$ contains a triple $(\tilde{s}, \perp, s)$. If it does, we output s as the answer to the oracle call,
- else, we browse the $\mathsf{G\text{-}List}$ and check for all triples $(\perp, u, s)$ whether $(\mathbf{pk_A}, \mathbf{pk_B}, uP_1, \tilde{s})$ is a valid Bilinear Diffie-Hellman quadruple. If it does, we give s as the answer,
- otherwise we pick at random $s \in \{0,1\}^{k_2}$, record $(\tilde{s}, \perp, s)$ in the $\mathsf{G\text{-}List}$, and output s as the answer to the oracle call.

We have $\Pr[\mathsf{Forge}_3] = \Pr[\mathsf{Forge}_2]$.

Game_4 We now simulate the signing oracle: for any m, whose signature is queried, we pick at random three elements $r \in \{0,1\}^{n_r}$, $s \in \{0,1\}^{k_2}$, $u \in [\![1, q-1]\!]$, and compute $Q = uP_1$.

- If the $\mathsf{H\text{-}List}$ includes a quadruple $(m, r, ?, ?)$ we abort the simulation, else we store (m, r, u, Q) in the $\mathsf{H\text{-}List}$,
- we browse the $\mathsf{G\text{-}List}$ and check for all triples $(\tilde{s}, \perp, ?)$ (resp. $(\perp, v, ?)$) whether $(\mathbf{pk_A}, \mathbf{pk_B}, uP_1, \tilde{s})$ is a valid Bilinear Diffie-Hellman quadruple (resp. wether $u = v$). If it does, we abort the simulation,
- otherwise, we record $(\perp, u, s)$ in the $\mathsf{G\text{-}List}$, and output (r, s).

Since there are at most $q_\mathcal{H} + q_\Sigma + 1$ messages queried to the random oracle $\mathcal{H}$ and $q_\mathcal{G} + q_\Sigma + 1$ messages queried to the random oracle $\mathcal{G}$, the new simulation aborts with probability at most $(q_\mathcal{H} + q_\Sigma + 1) \cdot 2^{-n_r} + (q_\mathcal{G} + q_\Sigma + 1) \cdot 2^{-k_1}$. Otherwise, this new oracle perfectly simulates the signature. Summing up for all signing queries, we obtain

$$|\Pr[\mathsf{Forge}_4] - \Pr[\mathsf{Forge}_3]| \leq \left(\frac{(q_\mathcal{H} + q_\Sigma + 1)}{2^{n_r}} + \frac{(q_\mathcal{G} + q_\Sigma + 1)}{2^{k_1}} \right) q_\Sigma$$

Game_5 In this game, we make the verifying oracle reject all couples message/signature $(m, (r, s))$ such that s has not been obtained from $\mathcal{G}$. As in Game_5 of the previous proof, we get $|\Pr[\mathsf{Forge}_5] - \Pr[\mathsf{Forge}_4]| \leq (q_\Upsilon + 1)2^{-k_2}$.

Game_6 In this game, we finally simulate the verifying oracle. For any couple message/signature $(m, (r, s))$, whose verification is queried, we check whether the $\mathsf{H\text{-}List}$ includes a quadruple $(m, r, ?, ?)$. If it does not, we reject the signature. Therefore the $\mathsf{H\text{-}List}$ includes a quadruple (m, r, u, Q), and we browse the $\mathsf{G\text{-}List}$: if it includes a triple $(\tilde{s}, \perp, s)$, we accept the signature if and only if $(\mathbf{pk_A}, \mathbf{pk_B}, Q, \tilde{s})$ is a valid Bilinear Diffie-Hellman quadruple; else the $\mathsf{G\text{-}List}$ includes a triple $(\perp, v, s)$ and we accept the signature if and only if $u = v$. As in Game_6 of the previous proof, we get

$$|\Pr[\mathsf{Forge}_6] - \Pr[\mathsf{Forge}_5]| \leq \frac{(q_\mathcal{G} + q_\Sigma + 1)(q_\Upsilon + 1)}{2^{k_1}} + \frac{q_\Sigma (q_\Upsilon + 1)}{2^{k_2}}.$$

When the game Game_6 terminates, outputting a valid message/signature $(m^\star, (r^\star, s^\star))$ pair, by definition of existential forgery, the $\mathsf{H\text{-}List}$ includes a quadruple $(m^\star, r^\star, u^\star, Q^\star)$ with $Q^\star = u^\star R_3$.
By the simulation $(\mathbf{pk_A}, \mathbf{pk_B}, Q^\star, \tilde{s}^\star)$ is a valid Bilinear Diffie-Hellman quadruple, and therefore $z = (\tilde{s}^\star)^{(u^\star)^{-1}}$ gives the solution to the GBDH problem instance (R_1, R_2, R_3), and we obtained the claimed bounds.

Theorem 4 (Anonymity of DVSBMH). *Let $\mathcal{G}en$ be a prime-order-BDH-parameter-generator, let f_1, f_2, $f_r : \mathbb{N} \to \mathbb{N}$ be three functions and let DVSBMH be the associated DVS scheme. For any PSI-CMA-adversary $\mathcal{A}$, in the random oracle model, against DVSBMH, with security parameter k which has advantage $\varepsilon = \mathbf{Adv}^{\mathsf{psi-cma}}_{\mathit{DVSBMH},\mathcal{A}}(k)$, running time t, and makes $q_{\mathcal{H}}$ and $q_{\mathcal{G}}$ queries to the random oracles, q_Σ queries to the signing oracles and q_Υ queries to the verifying oracle, there exists an adversary D for GBDH which has success $\varepsilon' = \mathbf{Succ}^{\mathsf{gbdh}}_{\mathcal{G}en,D}(k)$ running in time $t' \in \mathbb{N}$ such that*

$$\begin{cases} \varepsilon' \ \geq \ \dfrac{\varepsilon}{2} - \dfrac{q_\Sigma(q_{\mathcal{H}} + q_\Sigma + 1)}{2^{n_r}} - \dfrac{(q_{\mathcal{G}} + q_\Sigma + 1)}{2^{k_1}}(q_\Sigma + q_\Upsilon + 1) - \dfrac{(q_\Upsilon + 1)(q_\Sigma + 1)}{2^{k_2}} \\ t' \leq t + (q_{\mathcal{H}} + 2q_\Sigma + 1)\left(T_{Exp-\mathbb{G}_1} + O(1)\right) \\ \qquad\quad + (q_{\mathcal{G}} + q_\Sigma + 1)(q_{\mathcal{G}} + q_\Sigma + q_\Upsilon)\left(T_{DDH} + O(1)\right), \end{cases}$$

where $k_1 = f_1(k)$, $k_2 = f_2(k)$ and $n_r = f_r(k)$.

5 Conclusion

We designed an efficient construction for strong DVS based on *any* bilinear map (which is a variant of DVSBM from [17]), and clarified the property of anonymity of the signers. Unlike Steinfeld *et al.*, our construction can be instantiated with the discrete exponentiation. In this case, the unforgeability and the privacy of signer's identity are related to the Gap Diffie-Hellman problem, since the discrete logarithm in $\mathbb{G}_1$ is easy. This new scheme offers the best performance in terms of computational cost and signature length. The DVSBMH scheme built on the discrete exponentiation is closely bound to a Diffie-Hellman session key exchange. The general relationship between session key exchange and DVS seems to be an interesting topic for further research.

Ackowledgements We express our gratitude to Jacques Traoré, Pascal Paillier and Éric Reyssat for their helpful comments. Many thanks to Pierre and Laura for correcting some misprints and our broken english.

References

1. M. Bellare, A. Boldyreva, A. Desai, D. Pointcheval: Key-Privacy in Public-Key Encryption. Proc. of Asiacrypt'01, Springer LNCS Vol. 2248, 566-582 (2001)
2. M. Bellare, A. Desai, D. Pointcheval and P. Rogaway: Relations among Notions of Security for Public-Key Encryption Schemes. Proc of Crypto'98, Springer LNCS Vol. 1462, 162–177 (1998)

3. M. Bellare, P. Rogaway: Random Oracles are Practical: a Paradigm for Designing Efficient Protocols. Proc. of 1st ACM Conference on Computer and Communications Security, 62–73 (1993)
4. D. Boneh, M. Franklin: Identity-based Encryption from the Weil Pairing. SIAM J. Computing, 32(3), 586–615 (2003)
5. D. Boneh, B. Lynn, H. Shacham: Short Signatures from the Weil Pairing. Proc of Asiacrypt'01, Springer LNCS Vol. 2248, 514–532 (2001)
6. J. Camenisch: Efficient and Generalized Group Signatures. Proc of Eurocrypt'97, Springer LNCS Vol. 1233, 465–479 (1997)
7. D. Chaum: Private Signature and Proof Systems. United States Patent 5,493,614 (1996)
8. E.-J. Goh, S. Jarecki: A Signature Scheme as Secure as the Diffie-Hellman Problem. Proc of Eurocrypt'03, Springer LNCS Vol. 2656, 401–415 (2003)
9. S. Goldwasser, S. Micali, R. L. Rivest: A digital signature scheme secure against adaptative chosen-message attacks. SIAM J. of Computing, 17 (2) 281–308 (1988)
10. M. Jakobsson, K. Sako, R. Impagliazzo: Designated Verifier Proofs and their Applications. Proc. of Eurocrypt'96, Springer LNCS Vol. 1070, 142–154 (1996)
11. F. Laguillaumie, D. Vergnaud: Efficient and Provably Secure Designated Verifier Signature Schemes from Bilinear Maps. Crypto'03 rump session. Rapport de Recherche LMNO, 2003-25, 16 pages (2003)
12. F. Laguillaumie, D. Vergnaud: Designated Verifier Signatures: Anonymity and Efficient Construction from *any* Bilinear Map. Full version, IACR e-print.
13. T. Okamoto, D. Pointcheval: The Gap-Problems: a New Class of Problems for the Security of Cryptographic Schemes. Proc. of PKC'01 Springer LNCS Vol. 1992, 104–118 (2001)
14. R. L. Rivest, A. Shamir, Y. Tauman: How to Leak a Secret. Proc. of Asiacrypt'01, Springer LNCS Vol. 2248, 552–565 (2001)
15. S. Saeednia, S. Kremer, O. Markowitch: An Efficient Strong Designated Verifier Signature Scheme. Proc. of ICISC 2003, Springer LNCS Vol. 2836, 40–54 (2003)
16. V. Shoup: OAEP reconsidered. J. Cryptology, Vol. 15 (4), 223–249 (2002)
17. R. Steinfeld, L. Bull, H. Wang, J. Pieprzyk: Universal Designated Verifier Signatures. Proc. of Asiacrypt'03, Springer LNCS Vol. 2894, 523–542 (2003)
18. R. Steinfeld, H. Wang, J. Pieprzyk: Efficient Extension of Standard Schnorr/RSA signatures into Universal Designated-Verifier Signatures. Proc. of PKC'04, Springer LNCS Vol. 2947, 86–100 (2004)
19. W. Susilo, F. Zhang, Y. Mu: Identity-based Strong Designated Verifier Signatures Schemes. Proc. of ACISP'04, Springer LNCS Vol. 3108, 313–324 (2004)

A Review of other schemes

A.1 Privacy of signer's identity of SchDVS_1, SchDVS_2 and RSADVS

In [18], Steinfeld *et al.* proposed three Universal DVS schemes $\mathsf{SchUDVS}_1$, $\mathsf{SchUDVS}_2$ and $\mathsf{RSAUDVS}$. We refer the reader to [18], for the description of these schemes. The DVS schemes induced by $\mathsf{SchUDVS}_2$ and $\mathsf{RSAUDVS}$ do not satisfy the $\mathsf{PSI\text{-}CMA}$ security property. Indeed, the designated verifier secret key is not involved in the verifying algorithm. However it is easy to see that SchDVS_1, the DVS scheme induced by $\mathsf{SchUDVS}_1$, fulfills this property assuming the difficulty of the Decision Diffie-Hellman (DDH) problem:

Theorem 5 (Anonymity of SchDVS_1). *Let $\mathcal{A}$ be a PSI-CMA-adversary, in the random oracle model, against the SchDVS_1 scheme, with security parameter k. Assume that $\mathcal{A}$ has advantage $\varepsilon = \mathbf{Adv}^{\mathsf{psi-cma}}_{\mathsf{SchDVS}_1,\mathcal{A}}(k)$, running time t, and makes $q_{\mathcal{H}}$, q_{Σ}, q_{Υ} queries to the hash function $\mathcal{H}$, to the signing oracles and to the verifying oracle (respectively). Then there exist $\varepsilon' \in [0,1]$ and $t' \in \mathbb{N}$ verifying*

$$\begin{cases} \varepsilon' \geq \dfrac{\varepsilon}{2} - \dfrac{q_{\Upsilon} + q_{\mathcal{H}} q_{\Sigma} + 1}{2^k} \\ t' \leq t + (q_{\Sigma} + q_{\Upsilon})(3T_{Exp-\mathbb{G}} + O(1)) \end{cases}$$

such that the DDH problem can be solved with probability ε', within time t'.

Proof. Due to lack of space, the proof will be given in the full version of the paper [12].

A.2 Security of Saeednia, Kremer and Markowitch's scheme (SKM)

The unforgeability of the DVS scheme in [15] is only considered under a no-message attack which is not acceptable in terms of security requirements. By using the technique introduced in the proof of Theorem 3, we can prove the unforgeability of SKM's scheme against a chosen message attack:

Theorem 6 (Unforgeability of SKM signatures). *Let $\mathcal{A}$ be an EF-CMA-adversary against SKM's scheme with security parameter k, in the random oracle model, which produces an existential forgery with probability $\varepsilon = \mathbf{Succ}^{\mathsf{ef-cma}}_{SKM,\mathcal{A}}(k)$, within time t, making $q_{\mathcal{H}}$, q_{Σ} and q_{Υ} queries to the hash oracle, to the signing oracle and to the verifying oracle. Then there exist $\varepsilon' \in [0,1]$ and $t' \in \mathbb{N}$ verifying*

$$\begin{cases} \varepsilon' \geq \varepsilon - \dfrac{(q_{\mathcal{H}} + q_{\Sigma})q_{\Sigma} + q_{\Upsilon}}{2^k}, \\ t' \leq t + (q_{\Sigma} + q_{\Upsilon})\left(2T_{Exp-\mathbb{G}} + O(1)\right) + (q_{\mathcal{H}} + q_{\Sigma})(q_{\mathcal{H}} + q_{\Sigma} + q_{\Upsilon})\left(T_{DDH} + O(1)\right), \end{cases}$$

such that the Gap Diffie-Hellman (GDH) problem can be solved with probability ε', within time t'.

Theorem 7 (Anonymity of SKM signatures). *Let $\mathcal{A}$ be a PSI-CMA-adversary, in the random oracle model, against SKM's scheme, with security parameter k. Assume that $\mathcal{A}$ has advantage $\varepsilon = \mathbf{Adv}^{\mathsf{psi-cma}}_{SKM,\mathcal{A}}(k)$, running time t, and makes $q_{\mathcal{H}}$, q_{Σ}, q_{Υ} queries to the hash function $\mathcal{H}$, to the signing oracles and to the verifying oracle. Then there exist $\varepsilon' \in [0,1]$ and $t' \in \mathbb{N}$ verifying*

$$\begin{cases} \varepsilon' \geq \dfrac{\varepsilon}{2} - \dfrac{(q_{\mathcal{H}} + q_{\Sigma})q_{\Sigma} + q_{\Upsilon}}{2^k} \\ t' \leq t + (q_{\Sigma} + q_{\Upsilon})\left(2T_{Exp-\mathbb{G}} + O(1)\right) + (q_{\mathcal{H}} + q_{\Sigma})(q_{\mathcal{H}} + q_{\Sigma} + q_{\Upsilon})\left(T_{DDH} + O(1)\right) \end{cases}$$

such that GDH can be solved with probability ε', within time t'.

Proofs. They are straightfoward adaptations of the proof of Theorem 3. Due to lack of space, they will be omitted.