New Extensions of Pairing-based Signatures into Universal (Multi) Designated Verifier Signatures*

Damien Vergnaud

École normale supérieure – C.N.R.S. – I.N.R.I.A.
45 rue d'Ulm, 75230 Paris Cedex 05 – France

Abstract. The concept of universal designated verifier signatures was introduced by Steinfeld, Bull, Wang and Pieprzyk at Asiacrypt 2003. These signatures can be used as standard publicly verifiable digital signatures but have an additional functionality which allows any holder of a signature to designate the signature to any desired verifier. This designated verifier can check that the message was indeed signed, but is unable to convince anyone else of this fact. We propose new efficient constructions for pairing-based short signatures. Our first scheme is based on Boneh-Boyen signatures and its security can be analyzed in the standard security model. We prove its resistance to forgery assuming the hardness of the so-called strong Diffie-Hellman problem, under the knowledge-of-exponent assumption. The second scheme is compatible with the Boneh-Lynn-Shacham signatures and is proven unforgeable, in the random oracle model, under the assumption that the computational bilinear Diffie-Hellman problem is untractable. Both schemes are designed for devices with constrained computation capabilities since the signing and the designation procedure are pairing-free. Finally, we present extensions of these schemes in the multi-user setting proposed by Desmedt in 2003.

Keywords: pairing-based cryptography, designated verifier signature, security analysis

1 Introduction

Recently many *universal designated verifier signature* protocols have been proposed (*e.g.* [14, 20, 23]). The present paper focuses on the proposal of two new efficient constructions for pairing-based short signatures [4, 5] and on the security treatment of them. The resistance to forgery of the first scheme relies on the hardness of the strong Diffie-Hellman problem, under the *knowledge-of-exponent assumption*, in the standard security model, and the one of the second scheme relies, in the random oracle model, on the hardness of a new computational problem (not easier than the widely used computational bilinear Diffie-Hellman problem).

1.1 Related work.

Many cryptographic primitives have been proposed to limit the *self-authenticating* property of digital signatures. The primary one: *undeniable signatures* – introduced by Chaum and van Antwerpen in 1989 [6] – appeared to have some weaknesses. The concept of *designated verifier signatures* was introduced by Jakobsson, Sako and Impagliazzo [11] in order to repair their so-called *lie detector problem.* Designated verifier signatures are intended to a specific and unique designated verifier, who is the only one able to check their validity. Motivated by privacy issues associated with dissemination of signed digital certificates, Steinfeld, Bull, Wang and Pieprzyk [20] defined, in 2003, a new kind of signatures called *universal designated-verifier signatures* (UDVS). This primitive can function as a standard publicly-verifiable digital signature scheme but has an additional functionality which allows any holder of a signature to designate the signature to any verifier. Again, the designated-verifier can check that the message was signed by the signer, but is unable to convince anyone else of this fact. Designated verifier signatures

* This is the full version of "New Extensions of Pairing-based Signatures into Universal Designated Verifier Signatures" [22] presented at ICALP'06.

(universal or not) have found numerous applications in financial cryptography (*e.g.* call for tenders, electronic voting, electronic auction or distributed contract signing).

Steinfeld *et al.* proposed an efficient UDVS scheme constructed using any bilinear group-pair. In collaboration with Laguillaumie, we suggested in [14] a variant which significantly improves this protocol. Both schemes are compatible with the key-generation, signing and verifying algorithms of the Boneh-Lynn-Shacham [5] signature scheme (BLS). In [4], Boneh and Boyen proposed efficient pairing-based short signatures (BB) whose security can be analyzed in the standard security model. A UDVS scheme compatible with a variant of Boneh and Boyen's scheme has been proposed by Zhang, Furukawa and Imai [23].

1.2 Contributions of the paper.

The main contribution of the paper is to provide a new efficient UDVS protocol compatible with the *original* Boneh-Boyen scheme. The idea underlying our design relies on the flexibility of BB signatures and specifically on their good behaviour under scalar multiplication. The new scheme, that we call UDVS-BB, is unforgeable in the standard security model assuming the hardness of the strong Diffie-Hellman problem [4], under the knowledge-of-exponent assumption (KEA) [2, 8]. The protocol proposed by Zhang *et al.* is proven unforgeable assuming the hardness of the same algorithmic problem, but under an additional assumption (which is stronger than KEA). The security of UDVS-BB can also be proved under a well-defined (though *ad hoc*) computational problem without using any non-black-box assumption (such as KEA). The computational workload of UDVS-BB amounts to three exponentiations over bilinear groups for designating a signature and four pairing evaluations to verify it, and moreover, the length of the signatures is much smaller than the one of Zhang *et al.*'s signatures. Following the general paradigm from [13], this scheme is readily extended to produce universal multi designated verifier signatures (UMDVS) [16] that are verifiable in a non-interactive way. The multi-user scheme inherits the efficiency properties of UDVS-BB with the same signature size (which, in particular, does not grow with the number of verifiers).

Using the same design principle, we propose a new UDVS protocol compatible with the BLS signatures which is well-suited for devices with constrained computation capabilities and low bandwidth. Indeed the designation procedure of the signatures is pairing-free and the resulting size is comparable to the length of DSA signatures. The proof of security for this scheme, that we called UDVS-BLS, takes place in the random oracle model [3]: we show that this scheme is unforgeable with respect to a new computational assumption weaker than the widely used computational bilinear Diffie-Hellman assumption. In some cases [11, 14] it may be desirable that UDVSs provide a stronger notion of privacy. The scheme UDVS-BLS provides this security requirement assuming the hardness of the xyz-decisional co-Diffie Hellman problem. It is possible to extend this scheme into a UMDVS one.

2 Definitions

2.1 Notations

The set of n-bit strings is denoted by $\{0,1\}^n$ and the set of all finite binary strings is denoted by $\{0,1\}^*$. Let $\mathcal{A}$ be a probabilistic Turing machine running in polynomial time (a PPTM, for short), and let x be an input for $\mathcal{A}$. The probability space that assigns to a string σ the probability that $\mathcal{A}$, on input x, outputs σ is denoted by $\mathcal{A}(x)$. The support of $\mathcal{A}(x)$ is denoted by $\mathcal{A}[x]$. Given a probability space S, a PPTM that samples a random element according to S is denoted by $x \overset{R}{\leftarrow} S$. For a finite set X, $x \overset{R}{\leftarrow} X$ denotes a PPTM that samples a random element uniformly at random from X.

2.2 Universal designated verifier signatures

In this subsection, we recall the definitions of UDVS schemes and of their security requirements [13, 20].

Syntactic definition

Definition 1. *A* universal designated verifier signature scheme Σ *is an 8-tuple*

$$\Sigma = (\mathsf{Setup}, \mathsf{SKeyGen}, \mathsf{VKeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Designate}, \mathsf{Fake}, \mathsf{DVerify})$$

such that

- $(\mathsf{Setup}, \mathsf{SKeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *is a signature scheme:*
  - $\Sigma.\mathsf{Setup}$ *is a PPTM which takes an integer k as input. The output are the* public parameters $\mathcal{P}$. *k is called the* security parameter.
  - $\Sigma.\mathsf{SKeyGen}$ *is a PPTM which takes the public parameters as input. The output is a pair* $(\mathsf{sk_s}, \mathsf{pk_s})$ *where* $\mathsf{sk_s}$ *is called a* signing secret key *and* $\mathsf{pk_s}$ *a* signing public key.
  - $\Sigma.\mathsf{Sign}$ *is a PPTM which takes the public parameters, a message, and a signing secret key as inputs and outputs a bit string.*
  - $\Sigma.\mathsf{Verify}$ *is a PPTM which takes the public parameters, a message m, a bit string σ and a signing public key* $\mathsf{pk_s}$. *It outputs a bit. If the bit output is 1 then the bit string σ is said to be a* signature *on m for* $\mathsf{pk_s}$.
- $\Sigma.\mathsf{VKeyGen}$ *is a PPTM which takes the public parameters as input. The output is a pair* $(\mathsf{sk_v}, \mathsf{pk_v})$ *where* $\mathsf{sk_v}$ *is called a* verifying secret key *and* $\mathsf{pk_v}$ *a* verifying public key.
- $\Sigma.\mathsf{Designate}$ *is a PPTM which takes the public parameters, a message m, a signing public key* $\mathsf{pk_s}$*, a signature σ on m for* $\mathsf{pk_s}$ *and a verifying public key as inputs and outputs a bit string.*
- $\Sigma.\mathsf{Fake}$ *is a PPTM which takes the public parameters, a message, a signing public key and a verifying secret key as inputs and outputs a bit string.*
- $\Sigma.\mathsf{DVerify}$ *is a deterministic PPTM which takes the public parameters, a message m, a bit string τ, a signing public key* $\mathsf{pk_s}$*, a verifying public key* $\mathsf{pk_v}$ *and the matching verifying secret key* $\mathsf{sk_v}$ *as inputs. It outputs a bit. If the bit output is 1 then τ is said to be a* designated verifier signature *on m from* $\mathsf{pk_s}$ *to* $\mathsf{pk_v}$.

Σ must satisfies the following properties, for all $k \in \mathbb{N}$, all $\mathcal{P} \in \Sigma.\mathsf{Setup}[k]$, all $(\mathsf{pk_s}, \mathsf{sk_s}) \in \Sigma.\mathsf{SKeyGen}[\mathcal{P}]$, all $(\mathsf{pk_v}, \mathsf{sk_v}) \in \Sigma.\mathsf{VKeyGen}[\mathcal{P}]$ and all messages m:

- Correctness of Signature:

$$\forall \sigma \in \Sigma.\mathsf{Sign}[\mathcal{P}, m, \mathsf{sk_s}], \quad \Sigma.\mathsf{Verify}[\mathcal{P}, m, \sigma, \mathsf{pk_s}] = \{1\}.$$

- Correctness of Designation:

$$\forall \sigma \in \Sigma.\mathsf{Sign}[\mathcal{P}, m, \mathsf{sk_s}], \forall \tau \in \Sigma.\mathsf{Designate}[\mathcal{P}, m, \mathsf{pk_s}, \sigma, \mathsf{pk_v}], \\ \Sigma.\mathsf{DVerify}[\mathcal{P}, m, \tau, \mathsf{pk_s}, \mathsf{pk_v}, \mathsf{sk_v}] = \{1\}.$$

- Source Hiding:

$$\Sigma.\mathsf{Designate}(\mathcal{P}, m, \mathsf{pk_s}, \Sigma.\mathsf{Sign}(\mathcal{P}, m, \mathsf{sk_s}), \mathsf{pk_v}]) = \Sigma.\mathsf{Fake}(\mathcal{P}, m, \mathsf{pk_s}, \mathsf{sk_v}).$$

The correctness properties insure that a properly formed (designated verifier) signature is always accepted by the (designated) verifying algorithm. The source hiding property states that given a message m, a signing public key $\mathsf{pk_s}$, a verifying public key $\mathsf{pk_v}$ and a designated verifier signature τ on m from $\mathsf{pk_s}$ to $\mathsf{pk_v}$ it is (unconditionally) infeasible to determine if τ was produced by Σ.Designate or Σ.Fake.

Security requirements In this section, we state the definitions of *unforgeability* and *privacy of signer's identity* under a chosen message attack that were introduced in [14, 20]. In the following $\Sigma = (\mathsf{Setup}, \mathsf{SKeyGen}, \mathsf{VKeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Designate}, \mathsf{Fake}, \mathsf{DVerify})$ denotes a UDVS scheme.

Resistance to forgery. The accepted definition of security for signature schemes is existential unforgeability under adaptive chosen message attack [10]. The notion of UDVS-EF-CMA-security [14, 20] is a natural extension of this to the UDVS setting.

It is defined *via* a random experiment parameterized by a security parameter k. The experiment involves an adversarial user $\mathcal{A}$ and is as follows: first two public/secret key pairs for the signer and the verifier are generated by running the key generation algorithms. Then $\mathcal{A}$ engages in polynomially many runs of the signing oracle, the verifying oracle and – possibly – the random oracle, interleaved at its own choosing. Eventually, $\mathcal{A}$ outputs a pair $(m^\star, \tau^\star)$, such that $m^\star$ was never queried to the signing oracle, and it wins if the verifying oracle returns 1 when queried on this pair.

Definition 2. *Let $\mathcal{A}$ be a PPTM. We consider the following random experiments, where $k \in \mathbb{N}$ is a security parameter:*

```
$\boxed{\textit{Experiment } \mathbf{Exp}^{\mathsf{UDVS\text{-}EF\text{-}CMA}}_{\Sigma,\mathcal{A}}(k)}$
$\mathcal{L} \leftarrow \emptyset$
$\mathcal{P} \xleftarrow{R} \Sigma.\mathsf{Setup}(k)$
$(\mathsf{sk_s}, \mathsf{pk_s}) \xleftarrow{R} \Sigma.\mathsf{SKeyGen}(\mathcal{P})$ ; $(\mathsf{sk_v}, \mathsf{pk_v}) \xleftarrow{R} \Sigma.\mathsf{VKeyGen}(\mathcal{P})$
$(m^\star, \tau^\star) \xleftarrow{R} \mathcal{A}^{\mathfrak{S},\mathfrak{V}}(\mathcal{P}, \mathsf{pk_s}, \mathsf{pk_v})$
    $\mathfrak{S} : m \dashrightarrow \Sigma.\mathsf{Sign}(\mathcal{P}, m, \mathsf{sk_s}); \mathcal{L} \leftarrow \mathcal{L} \cup \{m\}$
    $\mathfrak{V} : (m, \tau) \dashrightarrow \Sigma.\mathsf{DVerify}(\mathcal{P}, m, \tau, \mathsf{pk_s}, \mathsf{pk_v}, \mathsf{sk_v})$
return 1 if $\Sigma.\mathsf{DVerify}(\mathcal{P}, m^\star, \tau^\star, \mathsf{pk_s}, \mathsf{pk_v}, \mathsf{sk_v}) = \{1\}$ and $m^\star \notin \mathcal{L}$
       0 otherwise.
```

Let $\tau, q_{\mathfrak{S}}, q_{\mathfrak{V}} \in \mathbb{N}^{\mathbb{N}}$, $\varepsilon \in [0,1]^{\mathbb{N}}$. We define the success *of $\mathcal{A}$* via

$$\mathbf{Succ}^{\mathsf{UDVS\text{-}EF\text{-}CMA}}_{\Sigma,\mathcal{A}}(k) = \Pr[\mathbf{Exp}^{\mathsf{UDVS\text{-}EF\text{-}CMA}}_{\Sigma,\mathcal{A}}(k) = 1].$$

1. *$\mathcal{A}$ is a $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{V}})$-UDVS-EF-CMA-*adversary *if for all $k \in \mathbb{N}$, the experiment $\mathbf{Exp}^{\mathsf{UDVS\text{-}EF\text{-}CMA}}_{\Sigma,\mathcal{A}}(k)$ ends in expected time less than $\tau(k)$ and in this experiment $\mathcal{A}$ makes at most $q_{\mathfrak{S}}(k)$ (*resp.*$q_{\mathfrak{V}}(k)$) queries to the oracle $\mathfrak{S}$ (*resp.*$\mathfrak{V}$).*
2. *Σ is $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{V}}, \varepsilon)$-UDVS-EF-CMA-*secure *if for any $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{V}})$-UDVS-EF-CMA-adversary $\mathcal{A}$ and any positive integer k, $\mathbf{Succ}^{\mathsf{UDVS\text{-}EF\text{-}CMA}}_{\Sigma,\mathcal{A}}(k) \leq \varepsilon(k)$.*

This definition does not capture that the adversary cannot generate a new signature on a previously signed message (the so-called *strong unforgeability*).

Privacy of signer's identity. As explained in [11], in some cases, it may be desirable that designated verifier signatures provide a stronger notion of privacy. More precisely, given a designated verifier signature and two potential signing public keys, it should be computationally infeasible for an eavesdropper, to determine under which of the two corresponding secret keys the signature was performed. The *privacy of signer's identity* (Ψ) property was formalized in [14] to capture this security notion.

We consider a $\mathsf{UDVS}\text{-}\Psi\text{-}\mathsf{CMA}$-adversary $\mathcal{A}$, which runs in two stages: in the `find` stage, it takes two signing public keys $\mathsf{pk_{s0}}$ and $\mathsf{pk_{s1}}$ and a verifying public key $\mathsf{pk_v}$, and outputs a message $m^\star$ together with some state information $\mathcal{I}^\star$. In the `guess` stage, it gets a challenge UDVS $\tau^\star$ formed 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 signing oracles $\mathfrak{S}$, to the verifying oracle $\mathfrak{V}$ and – possibly – to a random oracle. It is allowed to invoke them on any message with the restriction of not querying $m^\star$ from $\mathfrak{S}$ or $\mathfrak{V}$ in any stage.

Definition 3. *Let $\mathcal{A}$ be a PPTM. We consider the following random experiments, where $b \in \{0, 1\}$ and $k \in \mathbb{N}$ is a security parameter:*

```
$\boxed{\textit{Experiment } \mathbf{Exp}^{\mathsf{UDVS}\text{-}\Psi\text{-}\mathsf{CMA}-b}_{\Sigma,\mathcal{A}}(k)}$
$\mathcal{L} \leftarrow \emptyset$
$\mathcal{P} \xleftarrow{R} \Sigma.\mathit{Setup}(k)$
$(\mathsf{sk_{s0}}, \mathsf{pk_{s0}}) \xleftarrow{R} \Sigma.\mathit{SKeyGen}(\mathcal{P})$ ; $(\mathsf{sk_{s1}}, \mathsf{pk_{s1}}) \xleftarrow{R} \Sigma.\mathit{SKeyGen}(\mathcal{P})$,
$(\mathsf{sk_v}, \mathsf{pk_v}) \xleftarrow{R} \Sigma.\mathit{VKeyGen}(\mathcal{P})$
$(m^\star, \mathcal{I}^\star) \xleftarrow{R} \mathcal{A}^{\mathfrak{S},\mathfrak{V}}(\mathit{find}, \mathcal{P}, \mathsf{pk_{s0}}, \mathsf{pk_{s1}}, \mathsf{pk_v})$
    | $\mathfrak{S} : (m, i) \dashrightarrow \Sigma.\mathit{Sign}(\mathcal{P}, m, \mathsf{sk}_{\mathsf{s}i}); \mathcal{L} \leftarrow \mathcal{L} \cup \{m\}$
    | $\mathfrak{V} : (m, \tau, i) \dashrightarrow \Sigma.\mathit{DVerify}(\mathcal{P}, m, \tau, \mathsf{pk}_{\mathsf{s}i}, \mathsf{pk_v}, \mathsf{sk_v}); \mathcal{L} \leftarrow \mathcal{L} \cup \{m\}$
$\sigma^\star \xleftarrow{R} \Sigma.\mathit{Sign}(\mathcal{P}, m^\star, \mathsf{sk}_{\mathsf{s}b})$ ; $\tau^\star \xleftarrow{R} \Sigma.\mathit{Designate}(\mathcal{P}, m^\star, \mathsf{pk}_{\mathsf{s}b}, \sigma^\star, \mathsf{pk_v})$
$b^\star \xleftarrow{R} \mathcal{A}^{\mathfrak{S},\mathfrak{V}}(\mathit{guess}, \tau^\star, \mathcal{I}^\star)$
return 1 if $b = b^\star$ and $m^\star \notin \mathcal{L}$
       0 otherwise.
```

Let $\tau, q_{\mathfrak{S}}, q_{\mathfrak{V}} \in \mathbb{N}^{\mathbb{N}}$, $\varepsilon \in [0, 1]^{\mathbb{N}}$. We define the advantage *of $\mathcal{A}$* via

$$\mathbf{Adv}^{\mathsf{UDVS}\text{-}\Psi\text{-}\mathsf{CMA}}_{\Sigma,\mathcal{A}}(k) = \left| \Pr[\mathbf{Exp}^{\mathsf{UDVS}\text{-}\Psi\text{-}\mathsf{CMA}-0}_{\Sigma,\mathcal{A}}(k) = 1] - \Pr[\mathbf{Exp}^{\mathsf{UDVS}\text{-}\Psi\text{-}\mathsf{CMA}-1}_{\Sigma,\mathcal{A}}(k) = 1] \right|.$$

1. *$\mathcal{A}$ is a $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{V}})$-*$\mathsf{UDVS}\text{-}\Psi\text{-}\mathsf{CMA}$-adversary *if for all $k \in \mathbb{N}$, the experiment $\mathbf{Exp}^{\mathsf{UDVS}\text{-}\Psi\text{-}\mathsf{CMA}}_{\Sigma,\mathcal{A}}(k)$ ends in expected time less than $\tau(k)$ and in this experiment $\mathcal{A}$ makes at most $q_{\mathfrak{S}}(k)$ (*resp.*$q_{\mathfrak{V}}(k)$) queries to the oracle $\mathfrak{S}$ (*resp.*. $\mathfrak{V}$).*
2. *Σ is $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{V}}, \varepsilon)$-*$\mathsf{UDVS}\text{-}\Psi\text{-}\mathsf{CMA}$-secure *if for any $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{V}})$-$\mathsf{UDVS}\text{-}\Psi\text{-}\mathsf{CMA}$-adversary $\mathcal{A}$ and any positive integer k, $\mathbf{Adv}^{\mathsf{UDVS}\text{-}\Psi\text{-}\mathsf{CMA}}_{\Sigma,\mathcal{A}}(k) \leq \varepsilon(k)$.*

Remark 1. Recently, Lipmaa, Wang and Bao [15] have identified a new security requirement for designated verifier signatures, that they called the *non-delegatability.* This property captures the infeasibility for a signer to delegate her authentication capacity without revealing her private key. In spite of its interest, we do not consider this issue in the following. Indeed, in this paper, we focus on UDVSs, and it is quite easy to see that, if the underlying designated verifier signature scheme is non-delegatable then the basic signature scheme is universally forgeable under a chosen-message attack. In [21], we propose a new definition of non-delegatability for UDVS schemes and present some new schemes achieving this security requirement.

2.3 Bilinear maps and computational assumptions

The security of asymmetric cryptographic tools relies on assumptions about the hardness of certain algorithmic problems. Bilinear maps such as Weil or Tate pairing on elliptic curves and hyperelliptic curves have found various applications in cryptography (*e.g.* [4,5]). In the following, we review the definition of cryptographic bilinear maps and in order to highlight that our schemes apply to any instantiation of BLS and BB signatures, we do not pin down any particular generator, but instead parameterize definitions and security results by a choice of generator.

Definition 4. *A* prime-order-BDH-parameter-generator *is a PPTM that takes as input $k \in \mathbb{N}$ and outputs a tuple $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi)$ satisfying the following conditions:*

1. *q is a prime with $2^{k-1} < q < 2^k$;*
2. *$(\mathbb{G}_1, +)$, $(\mathbb{G}_2, +)$ and $(\mathbb{G}_3, \cdot)$ are groups of order q;*
3. *$\psi : \mathbb{G}_2 \longrightarrow \mathbb{G}_1$ is an isomorphism s.t. there exists a PPTM to compute ψ;*
4. *$\langle\cdot,\cdot\rangle : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_3$ satisfies the following properties:*
   - *(a) $\langle [a]Q, [b]R\rangle = \langle Q, R\rangle^{ab}$ for all $(Q,R) \in \mathbb{G}_1 \times \mathbb{G}_2$ and all $(a,b) \in \mathbb{Z}^2$;*
   - *(b) $\langle\cdot,\cdot\rangle$ is non degenerate (*i.e. *$\langle \psi(P), P\rangle \neq 1_{\mathbb{G}_3}$ for some $P \in \mathbb{G}_2$);*
   - *(c) there exists a PPTM to compute $\langle\cdot,\cdot\rangle$.*

Let $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi)$ be as above, $P_2 \in \mathbb{G}_2$ and let $P_1 = \psi(P_2)$. In margin to the classical Diffie-Hellman problems in the groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_3$, the introduction of bilinear maps in cryptography gives rise to new algorithmic problems [5,14]. For instance, to analyze the security of their signatures, Boneh and Boyen [4] introduced a new computational problem, on which relies also the unforgeability of our scheme UDVS-BB:

ℓ-Strong Diffie-Hellman (ℓ-SDH): let x be an integer smaller than q. Given an integer ℓ and $([x]P_2, \dots, [x^\ell]P_2) \in \mathbb{G}_2^\ell$, compute a pair $\big([(x+m)^{-1}]P_1, m\big)$ in $\mathbb{G}_1 \times [\![1, q-1]\!]$.

We will prove the unforgeability of UDVS-BB assuming the intractability of this problem under KEA and the one of a new ad-hoc problem (but not easier than the previous one under KEA):

$\mathcal{PR}_1(\ell)$: let x, y be two integers smaller than q.
Given $\ell \in \mathbb{N}$, $(m_1, \dots, m_\ell) \in [\![1, q]\!]^\ell$ and $([(x+m_1)^{-1}]P_2, \dots, [(x+m_\ell)^{-1}]P_2) \in \mathbb{G}_2^\ell$, compute a 4-tuple (m, R, S, T) in $\big([\![1, q-1]\!] \setminus \{m_1, \dots, m_{q_\mathfrak{S}(k)}\}\big) \times \mathbb{G}_1^3$ such that

$$\langle S, X + [m]P_2\rangle = \langle R, P_2\rangle \text{ and } \langle T, P_2\rangle = \langle R, Y\rangle. \qquad (1)$$

The unforgeability of UDVS-BLS relies also on a new algorithmic problem (but not easier than the widely used computational bilinear Diffie-Hellman problem):

$\mathcal{PR}_2$: let x, y, z be three integers smaller than q. Given $[x]P_1$, $[y]P_2$ and $[z]P_2$, compute a pair $(R, Q) \in \mathbb{G}_1 \times \mathbb{G}_2$ such that $\langle R, Q\rangle = \langle P_1, P_2\rangle^{xyz}$.

Its UDVS-$\varPsi$-CMA-security relies on the decisional variant of it that we denote $\mathcal{PR}_3$.

Definition 5. *Let $\ell \in \mathbb{N}^\mathbb{N}$ and $\mathcal{A}$ be a PPTM. We consider the following random experiments, where $k \in \mathbb{N}$ is a security parameter:*

```
$\boxed{\textit{Experiment } \mathbf{Exp}^{\mathcal{PR}_1(\ell)}_{\mathsf{Gen},\mathcal{A}}(k)}$
$\mathcal{P} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi) \overset{R}{\longleftarrow} \mathsf{Gen}(k)$
$P_2 \overset{R}{\longleftarrow} \mathbb{G}_2 \setminus \{\mathbb{O}_{\mathbb{G}_2}\}$ ; $(x, y) \overset{R}{\longleftarrow} [\![1, q-1]\!]^2$
$X \leftarrow [x]P_2, Y \leftarrow [y]P_2$
for $i$ from $1$ to $\ell(k)$ do
    $m_i \overset{R}{\longleftarrow} [\![1, q]\!]$ ; $R_i \leftarrow [(x+m_i)^{-1}]\psi(P_2)$
$(m, R, S, T) \overset{R}{\longleftarrow} \mathcal{A}(\mathcal{P}, P_2, X, Y, m_1, \dots,$
                              $m_{\ell(k)}, R_1, \dots, R_{\ell(k)})$
return $1$ if $(R, S, T) \in \mathbb{G}_1^3$
           and satisfies (1)
       $0$ otherwise
```

```
$\boxed{\textit{Experiment } \mathbf{Exp}^{\mathcal{PR}_2}_{\mathsf{Gen},\mathcal{A}}(k)}$
$\mathcal{P} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi) \overset{R}{\longleftarrow} \mathsf{Gen}(k)$
$P_2 \overset{R}{\longleftarrow} \mathbb{G}_2 \setminus \{\mathbb{O}_{\mathbb{G}_2}\}$ ; $(x, y, z) \overset{R}{\longleftarrow} [\![1, q-1]\!]^3$
$X \leftarrow [x]\psi(P_2), Y \leftarrow [y]P_2, Z \leftarrow [z]P_2$
$(R, Q) \overset{R}{\longleftarrow} \mathcal{A}(\mathcal{P}, P_2, X, Y, Z)$
return $1$ if $(R, Q) \in \mathbb{G}_1 \times \mathbb{G}_2$
           and $\langle R, Q\rangle = \langle \psi(P_2), P_2\rangle^{xyz}$
       $0$ otherwise
```

Let $\tau \in \mathbb{N}^{\mathbb{N}}$, $\varepsilon \in [0,1]^{\mathbb{N}}$ and let $i \in \{1,2\}$. We define the successes *of $\mathcal{A}$* via

$$\mathbf{Succ}^{\mathcal{PR}_i}_{\mathsf{Gen},\mathcal{A}}(k) = \Pr[\mathbf{Exp}^{\mathcal{PR}_i}_{\mathsf{Gen},\mathcal{A}}(k) = 1].$$

1. *$\mathcal{A}$ is a τ-$\mathcal{PR}_i$-*adversary *if for all positive integer k, the experiment $\mathbf{Exp}^{\mathcal{PR}_i}_{\mathsf{Gen},\mathcal{A}}(k)$ ends in expected time less than $\tau(k)$.*
2. Gen *is a (τ,ε)-$\mathcal{PR}_i$-*secure-generator *if for any τ-$\mathcal{PR}_i$-adversary $\mathcal{A}$ and any positive integer k, $\mathbf{Succ}^{\mathcal{PR}_i}_{\mathsf{Gen},\mathcal{A}}(k) \le \varepsilon(k)$.*

Definition 6. *Let $\mathcal{A}$ be a PPTM. We consider the following random experiments, where $k \in \mathbb{N}$ is a security parameter:*

```
$\boxed{\textit{Experiment } \mathbf{Exp}^{\mathcal{PR}_3\text{-}d}_{\mathsf{Gen},\mathcal{A}}(k)}$
$P_2 \xleftarrow{R} \mathbb{G}_2 \setminus \{\mathbb{O}_{\mathbb{G}_2}\}$ ; $(x,y,z,t) \xleftarrow{R} [\![1, q-1]\!]^4$
$X \leftarrow [x]\psi(P_2), Y \leftarrow [y]P_2, Z \leftarrow [z]P_2$
$\textbf{\textit{if}}\ d = 0\ \textbf{\textit{then}}\ (R,Q) \leftarrow ([xt]\psi(P_2), [yzt^{-1}]P_2)\ \textbf{\textit{otherwise}}\ (R,Q) \xleftarrow{R} \mathbb{G}_1 \times \mathbb{G}_2$
$b \xleftarrow{R} \mathcal{A}((q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), X, Y, Z, R, Q)$
$\textbf{\textit{return}}\ 1\ \textbf{\textit{if}}\ b = d$
          $0\ \textbf{\textit{otherwise}}$
```

Let $\tau \in \mathbb{N}^{\mathbb{N}}$, $\varepsilon \in [0,1]^{\mathbb{N}}$. We define the advantage *of $\mathcal{A}$* via

$$\mathbf{Adv}^{\mathcal{PR}_3}_{\mathsf{Gen},\mathcal{A}}(k) = \left| \Pr[\mathbf{Exp}^{\mathcal{PR}_3-0}_{\mathsf{Gen},\mathcal{A}}(k) = 1] - \Pr[\mathbf{Exp}^{\mathcal{PR}_3-1}_{\mathsf{Gen},\mathcal{A}}(k) = 1] \right|.$$

1. *$\mathcal{A}$ is a τ-$\mathcal{PR}_3$-*adversary *if for all positive integer k, the experiment $\mathbf{Exp}^{\mathcal{PR}_3}_{\mathsf{Gen},\mathcal{A}}(k)$ ends in expected time less than $\tau(k)$.*
2. Gen *is a (τ,ε)-$\mathcal{PR}_3$-*secure-generator *if for any τ-$\mathcal{PR}_3$-adversary $\mathcal{A}$ and any positive integer k, $\mathbf{Adv}^{\mathcal{PR}_3}_{\mathsf{Gen},\mathcal{A}}(k) \le \varepsilon(k)$.*

3 Description of the new schemes

In this section, we describe our new UDVS schemes. The general principle underlying the construction of UDVS-BB and UDVS-BLS is based on an elegant technique proposed by Damgård [8] and aimed at making public-key encryption scheme secure against (non-adaptive) chosen ciphertext attacks. We give in details the ideas underlying their design, since we are convinced that they may be of independent interest[1] (*e.g.* for the construction of new privacy-preserving signature schemes).

3.1 Design principle

Let $(\mathbb{G}, +)$ be a group of prime order q and let P be a generator of G. In 1991, Damgård [8] presented a simple variant of the Elgamal encryption scheme in $\mathbb{G}$. In his proposal, Alice publishes two public keys $A_1 = [a_1]P$ and $A_2 = [a_2]P$ and keeps secret their discrete logarithms a_1 and a_2. When Bob wants to privately send a message $M \in \mathbb{G}$ to Alice, he picks uniformly

[1] Since the publication of [22], Laguillaumie, Libert and Quisquater [12] have proposed new universal designated verifier signatures. The technique presented in this paper can be used to improve the efficiency of their schemes.

at random an integer $r \in [\![1, q-1]\!]$ and transmits the triple (Q_1, Q_2, C) where $Q_1 = [r]P$, $Q_2 = [r]A_1$ and $C = M + [r]A_2$. When she receives the ciphertext (Q_1, Q_2, C), Alice checks whether the equality $Q_2 = [a_1]Q_1$ holds: if it is the case, she retrieves the message M, as $M = C - [a_2]Q_1$, otherwise she rejects the ciphertext.

Damgård proved that if the DDH problem is hard in $\mathbb{G}$, then this scheme is semantically secure against (non-adaptive) chosen ciphertext attacks, if we assume the so-called knowledge-of-exponent assumption [2]. Intuitively this assumption states that, without the knowledge of a_1, the only way to generate couples $(Q_1, Q_2) \in \mathbb{G}^2$, verifying $Q_2 = [a_1]Q_1$, is to choose an integer $r \in [\![1, q-1]\!]$ and to compute $Q_1 = [r]P$ and $Q_2 = [r]A_1$.

There are many ways in which the formulation of KEA can be varied to capture this intuition that the only way to generate a Diffie-Hellman triple is to know the corresponding exponent [2, 8]. Usually, this is done by saying that for any PPTM outputting such a triple, there is an "extractor" than can return this exponent. For our purposes, it is necessary to allow the adversary to be randomized as in [1] (in that case, it is important that the extractor gets the coins ϖ of the adversary as an additional input, since otherwise the assumption is clearly false). We propose a similar definition suitable for bilinear structures.

Definition 7. *Let $\mathcal{A}$ and $\overline{\mathcal{A}}$ be two PPTM's. We consider the following random experiments, where $k \in \mathbb{N}$ is a security parameter:*

```
$\boxed{\textit{Experiment } \mathbf{Exp}^{\textsf{kea}}_{\textsf{Gen},\mathcal{A},\overline{\mathcal{A}}}(k)}$
$(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi) \xleftarrow{R} \textsf{Gen}(k)$
$P_2 \xleftarrow{R} \mathbb{G}_2 \setminus \{\mathbb{O}_{\mathbb{G}_2}\}$ ; $x \xleftarrow{R} [\![1, q-1]\!]$
$(R, S) \xleftarrow{R} \mathcal{A}_k((q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, [x]P_2)$
$r \leftarrow \overline{\mathcal{A}}_k((q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, [x]P_2; \varpi)$
return 1 if $(R, S) \in \mathbb{G}_1 \times \mathbb{G}_2$, $\psi(S) = [x]R$ and $R \neq [r]P_2$
       0 otherwise
```

We define the advantage *of $\mathcal{A}$ relative to $\overline{\mathcal{A}}$* via

$$\mathbf{Adv}^{\textsf{kea}}_{\textsf{Gen},\mathcal{A},\overline{\mathcal{A}}}(k) = \Pr\left[\mathbf{Exp}^{\textsf{kea}}_{\textsf{Gen},\mathcal{A},\overline{\mathcal{A}}}(k) = 1\right].$$

Let $\varepsilon \in [0, 1]^{\mathbb{N}}$

1. *$\overline{\mathcal{A}}$ is a ε-*kea-extractor *for $\mathcal{A}$ if for all positive integer k, $\mathbf{Adv}^{\textsf{kea}}_{\textsf{Gen},\mathcal{A},\overline{\mathcal{A}}}(k) \leq \varepsilon(k)$*
2. *We say that the knowledge-of-exponent assumption holds for* Gen *if there exists a PPTM $\overline{\mathcal{A}}$ such that for every PPTM $\mathcal{A}$, there exists a negligible function ε such that $\overline{\mathcal{A}}$ is a ε-*KEA-extractor *for $\mathcal{A}$.*

3.2 Description of the protocol UDVS-BB

Boneh-Boyen's signatures. In 2004, Boneh and Boyen [4] proposed a new application of bilinear structures to construct efficient short signatures. Their idea is to plug the message to be signed in the exponent and, in order to avoid trivial "homomorphic" forgeries, to do so in a non-linear way. For an entity whose private/public key pair is $(u, [u]P_2)$ in $[\![1, q-1]\!] \times \mathbb{G}_2$, the publication of the group element $\sigma = [(u+m)^{-1}]P_1$ seems to be a good mean to authenticate a message $m \in [\![1, q-1]\!]$. Indeed, the computation of σ for a given couple $(m, [u]P_2)$ is equivalent to the resolution of the so called co-CDH problem [5] and it seems to remain difficult even if the adversary is allowed to choose m and knows $[(u+m_1)^{-1}]P_1, \ldots, [(u+m_s)^{-1}]P_1$ in $\mathbb{G}$, for

```
Algorithm UDVS-BB.Setup
Input: $k$
Output: $\mathcal{P}$
$(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi) \xleftarrow{R} \mathsf{Gen}(k)$
$P_2 \xleftarrow{R} \mathbb{G}_2$ ; $P_1 \leftarrow \psi(P_2)$ $h \xleftarrow{R} \mathfrak{h}(q-1)$
$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_1, P_2, h]$
```

```
Algorithm UDVS-BB.Sign
Input: $\mathcal{P}$, $m$, $(u_a, v_a)$
Output: $\sigma$
$h \leftarrow h(m)$
repeat $r \xleftarrow{R} [\![1, q-1]\!]$
until $u_a + h + v_a r \neq 0 \mod q$
$S \leftarrow [(u_a + h + v_a r)^{-1}]P_1$
$\sigma \leftarrow (r, S)$
```

```
Algorithm UDVS-BB.VKeyGen
Input: $\mathcal{P}$
Output: $(\mathsf{sk_v}, \mathsf{pk_v})$
$u_b \xleftarrow{R} [\![1, q-1]\!]$
$\mathsf{sk_v} \leftarrow u_b$
$\mathsf{pk_v} \leftarrow [u_b]P_2$
```

```
Algorithm UDVS-BB.Fake
Input: $\mathcal{P}$, $m$, $u_b$, $(U_a, V_a)$
Output: $\tau$
$(r, t) \xleftarrow{R} [\![1, q-1]\!]^2$
$R \leftarrow [t]\,(\psi(U_a) + [h(m)]P_1 + [r]\psi(V_a))$
$Q_1 \leftarrow [t]P_1$, $Q_2 \leftarrow [u_b]R$, $Q_3 \leftarrow R$
$\tau \leftarrow (r, Q_1, Q_2, Q_3)$
```

```
Algorithm UDVS-BB.SKeyGen
Input: $\mathcal{P}$
Output: $(\mathsf{sk_s}, \mathsf{pk_s})$
$(u_a, v_a) \xleftarrow{R} [\![1, q-1]\!]^2$
$\mathsf{sk_v} \leftarrow (u_a, v_a)$
$\mathsf{pk_v} \leftarrow ([u_a]P_2, [v_a]P_2)$
```

```
Algorithm UDVS-BB.Verify
Input: $\mathcal{P}$, $m$, $(U_a, V_a)$, $(r, S)$
Output: $b$
$\alpha \leftarrow \langle S, U + [h(m)]P_2 + [r]V\rangle$
if $\alpha = \langle P_1, P_2\rangle$ then $b \leftarrow 1$ else $b \leftarrow 0$
```

```
Algorithm UDVS-BB.Designate
Input: $\mathcal{P}$, $m$, $(U_a, V_a)$, $U_b$, $(r, S)$
Output: $\tau$
$t \xleftarrow{R} [\![1, q-1]\!]$
$Q_1 \leftarrow [t]S$,  $Q_2 \leftarrow [t]\psi(U_b)$,
$Q_3 \leftarrow [t]P_1$
$\tau \leftarrow (r, Q_1, Q_2, Q_3)$
```

```
Algorithm UDVS-BB.DVerify
Input: $\mathcal{P}$, $m$, $u_b$, $(U_a, V_a)$,
       $(r, Q_1, Q_2, Q_3)$
Output: $b$
$\alpha_1 \leftarrow \langle Q_1, U_a + [h(m)]P_2 + [r]V_a\rangle$
$\alpha_2 \leftarrow \langle Q_3, P_2\rangle$
$\beta_1 \leftarrow \langle Q_3, [u_b]P_2\rangle$, $\beta_2 \leftarrow \langle Q_2, P_2\rangle$
if $\alpha_1 = \alpha_2 \wedge \beta_1 = \beta_2$ then $b \leftarrow 1$
                                  else $b \leftarrow 0$
```

Fig. 1. Description of the protocol $\mathsf{UDVS\text{-}BB}(\mathsf{Gen}, \mathfrak{h})$

some $m_i \in [\![1, q-1]\!] \setminus \{m\}$ $(i \in [\![1, s]\!])$. Boneh and Boyen have proved that this problem is not easier than the $(s+1)$-SDH problem. They also made the important remark that the use of a second pair of keys $(v, [v]P_2)$ in $[\![1, q-1]\!] \times \mathbb{G}$ enables to prove the unforgeability of the scheme under chosen message attacks, in the standard security model: they suggested to replace the signature σ by a couple $([(u + m + rv)^{-1}]P_1, r)$ where r is picked uniformly at random in $[\![1, q-1]\!]$. Finally, in order to be able to sign arbitrarily long messages, an hash function family $\mathfrak{h}$ is added to the public parameters, such that for every group order q output by Gen, $\mathfrak{h}(q)$ generates the description of a (collision resistant) hash function h which maps arbitrary long bit strings on elements from $[\![1, q-1]\!]$.

The scheme UDVS-BB. Let $\mathcal{P} = ((q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_1, P_2)$ and $\mathfrak{h}$ be as above, let $(U_a, V_a) \in \mathbb{G}_2$ be Alice's public key for BB signatures. The principle underlying the universal designated verifier signature scheme UDVS-BB is based on Damgård's idea. Let us suppose that Bob has published a public key $U_b = [u_b]P_2$ and that the pair $\sigma = (r, S)$ in $[\![1, q-1]\!] \times \mathbb{G}_1$ is a BB signature produced by Alice, on a message m. If Cindy wants to designate σ to Bob, she picks uniformly at random an integer $t \in [\![1, q-1]\!]$ and sets $Q_1 = [t]S$, $Q_2 = [t]U_b$ and $Q_3 = [t]P_2$. The quadruple $\tau = (r, Q_1, Q_2, Q_3)$ is the resulting designated verifier signature on m. The protocol UDVS-BB is described with all the details in figure 1. The following simple observations are intuitive arguments in favor of the security of the protocol.

1. Under KEA, the equality
$$\langle Q_3, U_b \rangle = \langle Q_2, P_2 \rangle \tag{2}$$
insures Bob that Cindy knows the value t such that $Q_2 = [t]U_b$ and $Q_3 = [t]P_2$.
2. If (2) is satisfied, Bob is convinced that Cindy knows the group element $S = [t^{-1}]Q_1$. The BB verification equality $\langle S, U_a + [h(m)]P_2 + [r]V_a \rangle = \langle P_1, P_2 \rangle$, holds if and only if the equality
$$\langle Q_1, U_a + [h(m)]P_2 + [r]V_a \rangle = \langle Q_3, P_2 \rangle \tag{3}$$
does. Therefore, if the equalities (2) and (3) are true, the quadruple τ proves to Bob that Alice has actually signed the message m.
3. However, this quadruple cannot convince anyone else, since it could have been produced by Bob himself. Indeed, if Bob samples uniformly at random $(r, \tilde{t})$ in $[\![1, q-1]\!]^2$ and computes the group elements:
$$Q_1 = [\tilde{t}]P_1, Q_2 = [u_b \tilde{t}]\left(U_a + [h(m)]P_2 + [r]V_a\right), Q_3 = [\tilde{t}]U_a + [\tilde{t} \cdot h(m)]P_2 + [\tilde{t} \cdot r]V_a,$$
he produces quadruples which verify (2) and (3) and follow the same distribution as those produced by Cindy (namely with $t \equiv_q \tilde{t}(u_a + h(m) + v_a r)$).

Remark 2. Given a UDVS produced by UDVS-BB, it is easy, by random scalar multiplication, to produce a new signature on *the same* message for the *same* public keys. It is admitted that weak forgery is no real threat whatsoever.

Remark 3. The computational workload of UDVS-BB.DVerify for the designated verifier can be reduced to only two pairing evaluations and one bilinear exponentiation thanks to the knowledge of u_b by checking that $Q_2 = [u_b]Q_3$ instead of $\beta_1 = \beta_2$.

Remark 4. In the algorithm UDVS-BB.Fake, the verifier's secret key u_b is used only to compute $Q_2 = [u_b]R$. Therefore, the signer as well as the verifier can delegate his authenticating capacity (without revealing the secret key) by publishing the elements $K_1 = [u_a \cdot u_b]P_1$ and $K_2 = [v_a \cdot u_b]P_1$ in $\mathbb{G}_2$. Indeed, the knowledge of (K_1, K_2) suffices to produce an UDVS $\tau = (r, Q_1, Q_2, Q_3)$ on a message m by picking uniformly at random $(r, t) \in [\![1, q-1]\!]^2$, and computing $Q_1 \leftarrow [t]P_1$, $Q_2 \leftarrow [t]K_1 + [t \cdot h(m)]\psi(U_b) + [t \cdot r]K_2$ and $Q_3 = [t]\psi(U_a) + [t \cdot h(m)]P_1 + [t \cdot r]\psi(V_a)$. Therefore, the scheme UDVS-BB is delegatable.

3.3 Description of the protocol UDVS-BLS

Boneh-Lynn-Shacham's signatures. In [5], Boneh *et al.* presented the signature scheme BLS that works in any bilinear cryptographic context. The scheme resembles the undeniable signature scheme proposed by Chaum and van Antwerpen [6] and can be seen as a variant of the FDH signature scheme [3]. The protocol BLS is efficient, produces short signatures (for carefully chosen parameters), and is unforgeable in the random oracle model assuming the intractability of the co-CDH problem.

The scheme UDVS-BLS. Let Gen be a prime-order-BDH-parameter-generator, let $f_r \in \mathbb{N}^{\mathbb{N}}$, and let $\mathfrak{H}$ be an hash function family such that for bilinear structure $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi)$ output by Gen, $\mathfrak{H}(\mathbb{G}_1)$ generates the description of an hash function $\mathcal{H}$ (modeled in the security analysis as a random oracle) which maps arbitrary long bit strings on elements from $\mathbb{G}_1$. Let BLS be the associated signature scheme; using the same approach, it is possible to construct a new UDVS scheme compatible with the BLS signatures. The protocol UDVS-BLS is described with all the details in figure 2.

```
Algorithm UDVS-BLS.Setup
Input: $k$
Output: $\mathcal{P}$
$(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi) \xleftarrow{R} \mathsf{Gen}(k)$
$P_2 \xleftarrow{R} \mathbb{G}_2$ ; $n_r \leftarrow f_r(k)$ ; $\mathcal{H} \xleftarrow{R} \mathfrak{H}(\mathbb{G}_1)$
$\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, n_r, \mathcal{H}]$
```

```
Algorithm UDVS-BLS.Sign
Input: $\mathcal{P}$, $m$, $u$
Output: $\sigma$
$r \xleftarrow{R} \{0,1\}^{n_r}$ ; $H \leftarrow \mathcal{H}(m, r)$
$S \leftarrow [u]H$ ; $\sigma \leftarrow (r, S)$
```

```
Algorithm UDVS-BLS.VKeyGen
Input: $\mathcal{P}$
Output: $(\mathsf{sk_v}, \mathsf{pk_v})$
$\mathsf{sk_v} = u_b \xleftarrow{R} [\![1, q-1]\!]$
$\mathsf{pk_v} = U_b \leftarrow [u_b]P_2$
```

```
Algorithm UDVS-BLS.Fake
Input: $\mathcal{P}$, $m$, $\mathsf{pk_s}$, $\mathsf{sk_v}$
Output: $\tau$
$r \xleftarrow{R} \{0,1\}^{n_r}$ ; $t \xleftarrow{R} [\![1, q-1]\!]$
$Q_1 \leftarrow [t^{-1}]H(m, r)$ ; $Q_2 \leftarrow [t \cdot \mathsf{sk_v}]\mathsf{pk_s}$
$\tau \leftarrow (r, Q_1, Q_2)$
```

```
Algorithm UDVS-BLS.SKeyGen
Input: $\mathcal{P}$
Output: $(\mathsf{sk_s}, \mathsf{pk_s})$
$\mathsf{sk_s} = u_a \xleftarrow{R} [\![1, q-1]\!]$
$\mathsf{pk_s} = U_a \leftarrow [u_a]P_2$
```

```
Algorithm UDVS-BLS.Verify
Input: $\mathcal{P}$, $m$, $U_a$, $(r, S)$
Output: $b$
$H \leftarrow \mathcal{H}(m, r)$
$s \leftarrow \langle H, U_a\rangle$
if $s = \langle S, P_2\rangle$ then $b \leftarrow 1$ else $b \leftarrow 0$
```

```
Algorithm UDVS-BLS.Designate
Input: $\mathcal{P}$, $m$, $\mathsf{pk_s}$, $(r, S)$, $\mathsf{pk_v}$
Output: $\tau$
$t \xleftarrow{R} [\![1, q-1]\!]$
$Q_1 \leftarrow [t]S$ ; $Q_2 \leftarrow [t^{-1}]\mathsf{pk_v}$
$\tau \leftarrow (r, Q_1, Q_2)$
```

```
Algorithm UDVS-BLS.DVerify
Input: $\mathcal{P}$, $m$, $\mathsf{pk_s}$, $(r, Q_1, Q_2)$, $\mathsf{sk_v}$
Output: $b$
$H \leftarrow H(m, r)$ $s \leftarrow \langle [\mathsf{sk_v}]H, \mathsf{pk_s}\rangle$
if $s = \langle Q_1, Q_2\rangle$ then $b \leftarrow 1$
                    else $b \leftarrow 0$
```

Fig. 2. Description of the protocol $\mathsf{UDVS\text{-}BLS}(\mathsf{Gen}, f_r, \mathfrak{H})$

Let $k \in \mathbb{N}$, let $\mathcal{P} = ((q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, \mathcal{H})$ be some output of $\mathsf{BLS.Setup}(k)$ and let $U_a = [u_a]P_2$ (*resp.* $U_b = [u_b]P_2$) be Alice's (*resp.* Bob's) public key output by $\mathsf{BLS.KeyGen}(\mathcal{P})$. BLS signatures are elements $S = [u_a]H \in \mathbb{G}_1$, where the group element H is the hash value of the signed message m and (potentially) some random salt (of size $n_r = f_r(k)$). The discrete logarithm of H is unknown to all users, therefore, whence the signature S is randomized as above: $Q_1 = [t]S$ for some $t \in [\![1, q-1]\!]$, it suffices to reveal the element $Q_2 = [t^{-1}]U_b$ to prove, in a non-transferable way, to Bob that Alice actually signed the message m. The tuple $(P_2, U_a, U_b, H, \langle Q_1, Q_2\rangle)$ is indeed a bilinear Diffie-Hellman tuple which could have been produced by using secret information from Alice or Bob, but not otherwise under the assumption that the computational bilinear Diffie-Hellman problem problem is intractable.

Remark 5. The protocol UDVS-BLS is delegatable [15]. Indeed, in the algorithm UDVS-BLS.Fake, the secret key u_b from the designated verifier is only used to compute the element $Q_2 = [t{\cdot}u_b]U_a \in \mathbb{G}_2$ and the signer as well as the verifier can delegate their authenticating capability (without disclosing their secret key) by publishing the element $[u_a \cdot u_b]P_2$.

4 Security results

In this section, we state the security properties of our schemes.

4.1 Unforgeability of the scheme UDVS-BB

The theorem below states that the protocol $\mathsf{UDVS\text{-}BB}(\mathsf{Gen}, \mathfrak{h})$ is UDVS-EF-CMA-secure assuming the KEA assumption, the collision-resistance of $\mathfrak{h}$ and the intractability of the problem ℓ-SDH in

Gen, for all polynomial $\ell \in \mathbb{N}^{\mathbb{N}}$. Since KEA is a somewhat strange and impractical assumption, it would be better if we could do without it, as it has been recently done by Gjøsteen [9] for Damgård's encryption scheme. In the following theorem, we prove the unforgeability of $\mathsf{UDVS\text{-}BB}$ to $\mathcal{PR}_1(\ell)$ without KEA. Finally, since the protocol $\mathsf{UDVS\text{-}BB}$ is publicly verifiable, we consider only $\mathsf{UDVS\text{-}EF\text{-}CMA}$-attackers that do not make queries to the verifying oracle $\mathfrak{V}$.

Theorem 1. *Let* Gen *be a prime-order-BDH-generator and* $\mathfrak{h}$ *be an hash-function family of codomain indexed by the orders of groups generated by* Gen.

1. *If the scheme* $\mathsf{BB}(\mathsf{Gen}, \mathfrak{h})$ *is* $\mathsf{EF\text{-}CMA}$*-secure against polynomial adversaries, then under the KEA assumption in* Gen, *the scheme* $\mathsf{UDVS\text{-}BB}(\mathsf{Gen}, \mathfrak{h})$ *is* $\mathsf{UDVS\text{-}EF\text{-}CMA}$*-secure against polynomial adversaries.*
2. *If for all polynomial* ℓ, Gen *is* ℓ-SDH*-secure against polynomial adversaries and if* $\mathfrak{h}$ *is an hash-function collision-resistant against polynomial adversaries then, under the KEA assumption in* Gen, *the protocol* $\mathsf{UDVS\text{-}BB}(\mathsf{Gen}, \mathfrak{H})$ *is* $\mathsf{UDVS\text{-}EF\text{-}CMA}$*-secure against polynomial adversaries.*
3. *Let* $(\tau, q_{\mathfrak{S}}) \in \mathcal{F}(\mathbb{N}, \mathbb{N})^2$ *and* $\mathcal{A}$ *be a* $(\tau, q_{\mathfrak{S}}, 0)$-$\mathsf{UDVS\text{-}EF\text{-}CMA}$*-adversary against* $\mathsf{UDVS\text{-}BB}(\mathsf{Gen}, \mathfrak{h})$. *There exist* $\tau', \tau'' \in \mathcal{F}(\mathbb{N}, \mathbb{N})$ *verifying,*
$$\tau' = \tau + q_{\mathfrak{S}} \cdot (T_{exp}(\mathbb{G}_1) + O(1)) \text{ and } \tau'' = \tau + O(1),$$
a τ'-$\mathcal{PR}_1(q_{\mathfrak{S}})$*-adversary* $\mathcal{B}$ *against* Gen *and a* τ''-$\mathsf{Collision}$*-adversary* $\mathcal{C}$ *against* $\mathfrak{h}$ *such that,*
$$2 \cdot \mathbf{Succ}^{\mathcal{PR}_1(q_{\mathfrak{S}})}_{\mathsf{Gen},\mathcal{B}} + \mathbf{Succ}^{\mathsf{Collision}}_{\mathfrak{h},\mathcal{C}} \geq \mathbf{Succ}^{\mathsf{UDVS\text{-}EF\text{-}CMA}}_{\mathsf{UDVS\text{-}BB},\mathcal{A}}.$$

Proof.

1. The algorithm $\mathcal{B}$ which try to forge a signature BB, takes as input some public parameters $\mathcal{P}$ and a signing public key $\mathsf{pk_s}$. It computes a verifying public key $U_b = [u_b]P_2$ by running the algorithm $\mathsf{UDVS\text{-}BB.VKeyGen}'(\mathcal{P})$ and then executes the algorithm $\mathcal{A}$ on the entries $\mathcal{P}$, $\mathsf{pk_s}$ and U_b. It forwards $\mathcal{A}$'s signature queries to its own signing oracle and the simulation of the verifying oracles is straightforward since the protocol $\mathsf{UDVS\text{-}BB}$ is publicly verifiable.

   Let us denote $\mathcal{A}'$ the algorithm whose execution is identical to the one of $\mathcal{A}$, but which returns the pair (Q_3, Q_2), when $\mathcal{A}$ returns $\tau^\star = (r, Q_1, Q_2, Q_3)$. If $\mathcal{A}$'s output is a valid forgery, then the 4-tuple $(\psi(P_2), [u_b]\psi(P_2), Q_3, Q_2)$ is a valid Diffie-Hellman 4-tuple. Assuming KEA, there exists $\overline{\mathcal{A}'}$ which taken as inputs $\mathcal{A}'$'s random tape and entries, outputs $t \in [\![1, q-1]\!]$ such that $Q_3 = [t]P_2$ et $Q_2 = [t]U_b$ with a probability negligibly close to the success of $\mathcal{A}$.

   $\mathcal{B}$ run the algorithm $\overline{\mathcal{A}'}$ to get this value t and outputs the pair $\sigma^\star = (r, [t^{-1}]Q_1)$ which is a valid forgery for the scheme BB if τ is a valid forgery and $Q_3 = [t]P_2$. The probability of success of $\mathcal{B}$ is therefore negligibly close to the one of $\mathcal{A}$ and its running time is polynomial.
2. It is a simple consequence of the first part of the theorem and the security theorem from [4].
3. Let Gen be a prime-order-BDH-generator, $q_{\mathfrak{S}}, \tau \in \mathcal{F}(\mathbb{N}, \mathbb{N})$ and $\mathcal{A}$ be a $(\tau, q_{\mathfrak{S}}, 0)$-$\mathsf{UDVS\text{-}EF\text{-}CMA}$-adversary against $\mathsf{UDVS\text{-}BB}(\mathsf{Gen})$. It is readily seen that $\mathcal{A}$ can be converted into an attacker for the simplified scheme defined without the hash function $\mathfrak{h}$ or into an attacker $\mathcal{C}$ against the collision resistance of $\mathfrak{h}$. For the sake of simplicity, we will assume that the scheme works directly on messages $m \in [\![1, q]\!]$.

   We will construct an algorithm $\mathcal{B}$ which takes as inputs $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi)$ generated by $\mathsf{Gen}(k)$, a vector $(m_1, \ldots, m_{q_{\mathfrak{S}}(k)}) \in [\![1, q]\!]^{q_{\mathfrak{S}}(k)}$, $(P_2, X, Y) \in \mathbb{G}_2^3$ and $(R_1, \ldots, R_{q_{\mathfrak{S}}(k)}) \in$

$\mathbb{G}_1^{q_{\mathfrak{S}}(k)}$ satisfying $R_i = [(x+m_i)^{-1}]P_1$ for all $i \in [\![1, q_{\mathfrak{S}}(k)]\!]$ with $P_1 = \psi(P_2)$ et $X = [x]P_2$, outputs a 4-tuple

$$(m, R, S, T) \in \big([\![1, q-1]\!] \setminus \{m_1, \ldots, m_{q_{\mathfrak{S}}(k)}\}\big) \times \mathbb{G}_1^3$$

which satisfies (1).
Our method of proof is inspired by Shoup [19]: we define a sequence of games $\mathsf{Exp}_1, \ldots, \mathsf{Exp}_4$ starting from the actual UDVS-EF-CMA-adversary $\mathcal{A}$ and modify it step by step, until we reach a final game whose success probability has an upper bound related to solving the $\mathcal{PR}_1(q_{\mathfrak{S}})$ problem. All the games operate on the same underlying probability space: the public and private keys of the signature scheme and the coin tosses of $\mathcal{A}$.

Exp_1 $\mathcal{B}_1$ plays the role of the challenger in the experiment $\mathbf{Exp}^{\mathsf{UDVS\text{-}EF\text{-}CMA}}_{\mathsf{UDVS\text{-}BB},\mathcal{A}}$ of the definition 2:

```
Initialization  $k$
    $\mathcal{P} \overset{R}{\longleftarrow} \mathsf{UDVS\text{-}BB.Setup}(k)$,
    $(\mathsf{sk_s}, \mathsf{pk_s}) \overset{R}{\longleftarrow} \mathsf{UDVS\text{-}BB.SKeyGen}(\mathcal{P})$, $(\mathsf{sk_c}, \mathsf{pk_c}) \overset{R}{\longleftarrow} \mathsf{UDVS\text{-}BB.VKeyGen}(\mathcal{P})$
    run $\mathcal{A}(\mathcal{P}, \mathsf{pk_s}, \mathsf{pk_c}) \rightsquigarrow (m^\star, \tau^\star)$.
Simulation of the oracles
    • $\mathfrak{S}(m) \leftrightsquigarrow \mathsf{UDVS\text{-}BB.Sign}(\mathcal{P}, m, \mathsf{sk_s})$.
```

In the random experiments Exp_i, for $i \in [\![1, 5]\!]$, we denote by $\mathsf{F_i}$, the event "$m^\star \notin \mathcal{Q}_{\mathfrak{S}}$ and $\mathsf{UDVS\text{-}BB.DVerify}(\mathcal{P}, m^\star, \mathsf{pk_s}, \tau^\star, \mathsf{pk_c}, \mathsf{sk_c}) = 1$". By definition, we have $\Pr[\mathsf{F_1}] = \mathbf{Succ}^{\mathsf{UDVS\text{-}EF\text{-}CMA}}_{\mathsf{UDVS\text{-}BB},\mathcal{A}}(k)$.

Exp_2 $\mathcal{B}_2$ modify the previous simulation by inserting the bilinear structure underlying the instance of the problem $\mathcal{PR}_1(q_{\mathfrak{S}})$ to solve in the public parameters $\mathcal{P}$.

```
Initialization  $k$
    $\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle \cdot, \cdot \rangle, \psi), P_1, P_2]$,
    $(u_a, v_a, u_b) \overset{R}{\longleftarrow} [\![1, q-1]\!]^3$, $(U_a, V_a, U_b) \leftarrow ([u_a]P_2, [v_a]P_2, [u_b]P_2)$
    run $\mathcal{A}(\mathcal{P}, (U_a, V_a), U_b) \rightsquigarrow (m^\star, \tau^\star)$.
Simulation of the oracles
    • $\mathfrak{S}(m) \leftrightsquigarrow \mathsf{UDVS\text{-}BB.Sign}(\mathcal{P}, m, (u_a, v_a))$.
```

The distribution of $\mathcal{A}$'s entries is unchanged and we have $\Pr[\mathsf{F_2}] = \Pr[\mathsf{F_1}]$.

Exp_3 The algorithm $\mathcal{B}_3$ precomputes the signatures given to the adversary $\mathcal{A}$ and then uses his knowledge of the secret key u_a or v_a in the chameleon hash function to answer $\mathcal{A}$'s signature queries. The algorithm $\mathcal{B}_3$ distinguishes two types of forgers:

Type 0: the forgers which
(a) either make a signature query on m such that $m = -u_a$;
(b) or return a forgery $(m^\star, \tau^\star)$ with $\tau^\star = (r^\star, Q_1^\star, Q_2^\star, Q_3^\star)$, such that $m^\star + v_a r^\star \notin \{m_1, \ldots, m_{q_{\mathfrak{S}}(k)}\}$.

Type 1: the other forgers, namely those which
(a) do not make a signature query on m such that $m = -u_a$;
(b) and return a forgery $(m^\star, \tau^\star)$ with $\tau^\star = (r^\star, Q_1^\star, Q_2^\star, Q_3^\star)$, such that $m^\star + v_a r^\star = m_i$ for some $i \in [\![1, q_{\mathfrak{S}}(k)]\!]$.

The adversary $\mathcal{A}$ is necessarily of one of these two types and the algorithm $\mathcal{B}_4$ picks uniformly at random a bit $\beta \in \{0, 1\}$. This algorithm will be able (at the end of the simulation) to solve the $\mathcal{PR}_1(q_{\mathfrak{S}})$ problem if the adversary $\mathcal{A}$ is of type β.

```
Initialization  $k$
    $\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_1, P_2]$, $c \leftarrow 1$, $\ell \leftarrow 0$, $\beta \xleftarrow{R} \{0,1\}$
    $(u_a, v_a, u_b) \xleftarrow{R} [\![1, q-1]\!]^3$ ; $(U_a, V_a, U_b) \leftarrow ([u_a]P_2, [v_a]P_2, [u_b]P_2)$
    $(h_1, \ldots, h_{q_{\mathfrak{S}}(k)}) \xleftarrow{R} [\![1, q]\!]^{q_{\mathfrak{S}}(k)}$
    for $i$ from $1$ to $q_{\mathfrak{S}}(k)$ do
        if $\beta = 0$ then $T_i \leftarrow [(u_a + h_i)^{-1}]P_1$ else $T_i \leftarrow [(v_a + h_i)^{-1}]P_1$
    run $\mathcal{A}(\mathcal{P}, (U_a, V_a), U_b) \rightsquigarrow (m^\star, \tau^\star)$.
Simulation of the oracles
    • $\mathfrak{S}(m)$: if $\beta = 0$ and $[m]P_2 = -U_a$ then $\ell \leftarrow -m \mod q$
              if $\beta = 0$ then $r \leftarrow (h_c - m) \cdot v_a^{-1} \mod q$, $S \leftarrow T_c$
                    else $r \leftarrow (u_a + m) \cdot h_c^{-1} \mod q$, $S \leftarrow [r^{-1}]T_c$
              if $r = 0$ then return $\perp$ else $c \leftarrow c + 1$, return $(r, S)$.
```

In both cases, the signatures produced by $\mathcal{B}_3$ are perfectly distributed. We have indeed $\langle S, U_a + [m]P_2 + [r]V_a\rangle = \langle T_c, U_a + [h_c]P_2\rangle = \langle P_1, P_2\rangle$, for $\beta = 0$ and $\langle S, U_a + [m]P_2 + [r]V_a\rangle = \langle [r^{-1}]T_c, [r \cdot h_c]P_2 + [r]V_a\rangle = \langle P_1, P_2\rangle$ for $\beta = 1$.
In the random experiment Exp_i, let us denote for $i \in \{3, 4\}$, $\mathsf{T_i}$ the event "$\mathcal{A}$ is of type β". The algorithm $\mathcal{B}_3$ aborts the simulation only if $\mathcal{A}$ is of type $1-\beta$. Therefore, we have $\Pr[\mathsf{F}_3|\mathsf{T}_3] = \Pr[\mathsf{F}_2]$.

Exp_4 $\mathcal{B}_4$ replace in the following the public keys given as inputs to $\mathcal{A}$ and the precomputed signatures (h_i, S_i) by elements coming from the instance of the problem to solve.

```
Initialization  $k$
    $\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_1, P_2]$, $c \leftarrow 1$, $\ell \leftarrow 0$, $\beta \xleftarrow{R} \{0,1\}$
    if $\beta = 0$ then $(U_a, U_b) \leftarrow (X, Y)$, $v_a \xleftarrow{R} [\![1, q-1]\!]$, $V_a \leftarrow [u_b]P_2$
            else $(V_a, U_b) \leftarrow (X, Y)$, $u_a \xleftarrow{R} [\![1, q-1]\!]$, $U_a \leftarrow [u_b]P_2$
    $(h_1, \ldots, h_{q_{\mathfrak{S}}(k)}) \leftarrow (m_1, \ldots, m_{q_{\mathfrak{S}}(k)})$
    $(S_1, \ldots, S_{q_{\mathfrak{S}}(k)}) \leftarrow (R_1, \ldots, R_{q_{\mathfrak{S}}(k)})$
    run $\mathcal{A}(\mathcal{P}, (U_a, V_a), U_b) \rightsquigarrow (m^\star, \tau^\star)$.
```

In the random experiment Exp_3, if $\beta = 0$ (*resp.* if $\beta = 1$) the knowledge of (u_a, u_b, v_b) (*resp.* of (v_a, u_b, v_b)) is not necessary to answer $\mathcal{A}$'s signature queries. Hence, $\mathcal{B}_4$ can still answer $\mathcal{A}$'s queries and since the distribution of the public keys and the precomputed signatures is unchanged, we get

$$\Pr[\mathsf{F}_4|\mathsf{T}_4] = \Pr[\mathsf{F}_3|\mathsf{T}_3].$$

Eventually, when $\mathcal{A}$ returns the pair $(m^\star, \tau^\star)$, with $\tau^\star = (r^\star, Q_1^\star, Q_2^\star, Q_3^\star)$, the algorithm $\mathcal{B}$ can solve the instance of the problem $\mathcal{PR}_1(q_{\mathfrak{S}}(k))$:

– if $\mathcal{A}$ is of type 1 and returns a forgery on a message $m^\star$ satisfying the relation $(m_i - m^\star) \cdot (r^\star)^{-1} = x \mod q$ or if $\mathcal{A}$ is of type 0 and has made a signature query on a message m such that $m = -x$, then $\mathcal{B}_4$ can retrieve the discrete logarithm of X in base P_2 and it can trivially produce a triple (R, S, T) verifying the equality (1);
– otherwise, $\mathcal{B}$ computes $m = m^\star + r^\star v_a \mod q$ and stops its execution by outputting the triple $(m, R, S, T) \in [\![1, q-1]\!] \times \mathbb{G}_2 \times \mathbb{G}_1 \times \mathbb{G}_3$ where $m = m^\star + r^\star v_a \mod q$ $R = Q_3^\star$, $S = Q_1^\star$ and $T = Q_2^\star$.

```
End of $\mathcal{B} = \mathcal{B}_4$'s execution  $(m^\star, \tau^\star)$
    $(r^\star, Q_1^\star, Q_2^\star, Q_3^\star) \leftarrow \tau^\star$
    if $\beta = 1$ and $\exists m_i, [(m_i - m^\star) \cdot (r^\star)^{-1}]P_2 = U_a$ do $\ell \leftarrow (m_i - m^\star) \cdot (r^\star)^{-1}$
    if $\ell \neq 0$ then $m \xleftarrow{R} [\![1, q-1]\!] \setminus \{m_1, \ldots, m_{q_{\mathfrak{S}}(k)}\}$ ; $r \xleftarrow{R} [\![1, q-1]\!]$
                $R \leftarrow [r]P_1$ ; $S \leftarrow [r(\ell + m)^{-1}]P_1$ ; $T \leftarrow [r]\psi(Y)$
    if $\beta = \ell = 0$ then $m \leftarrow m^\star + r^\star v_a \mod q$
                $R \leftarrow Q_3^\star$ ; $S \leftarrow Q_1^\star$ ; $T \leftarrow Q_2^\star$
    return $(m, R, S, T)$.
```

Clearly, if the event $\mathsf{F}_4 \cap \mathsf{T}_4$ occurs, the algorithm $\mathcal{B} = \mathcal{B}_4$ returns a 4-tuple (m, R, S, T) which satisfies (1) and since $\Pr[\mathsf{T}_4] = 1/2$, we get

$$2 \cdot \mathbf{Succ}^{\mathcal{PR}_1(q_{\mathfrak{S}})}_{\mathsf{Gen},\mathcal{B}} + \mathbf{Succ}^{\mathsf{Collision}}_{\mathfrak{h},\mathcal{C}} \geq \mathbf{Succ}^{\mathsf{UDVS\text{-}EF\text{-}CMA}}_{\mathsf{UDVS\text{-}BB},\mathcal{A}}.$$

The algorithm $\mathcal{B}$ runs in time less than $\tau(k) + q_{\mathfrak{S}}(k)(T_{\exp}(\mathbb{G}_1) + O(1))$, which concludes the proof.

4.2 Unforgeability and anonymity of the scheme UDVS-BLS

We prove (in the random oracle model) that UDVS-BLS is UDVS-EF-CMA-secure under the assumption that the problem $\mathcal{PR}_2$ is intractable in Gen. It is worth noting that this problem is at least as hard as the computational bilinear Diffie-Hellman problem underlying the schemes from [14, 20]. We prove also (again in the random oracle model) that the protocol UDVS-BLS is UDVS-Ψ-CMA-secure under the assumption that the decisional variant of this problem is intractable in Gen.

Theorem 2. *Let $f_r \in \mathcal{F}(\mathbb{N}, \mathbb{N})$ and let* Gen *be a prime-order-BDH-generator. Let $(q_{\mathfrak{S}}, q_{\mathfrak{V}}, q_{\mathfrak{H}}, \tau) \in \mathcal{F}(\mathbb{N}, \mathbb{N})^4$.*

1. *For all $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{V}})$-*UDVS-EF-CMA*-adversary $\mathcal{A}$ against* UDVS-BLS(Gen, f_r, $\mathcal{O}_{\mathfrak{H}}$) *where $\mathcal{O}_{\mathfrak{H}}$ is a $q_{\mathfrak{H}}$-random oracle, there exists $\tau' \in \mathcal{F}(\mathbb{N}, \mathbb{N})$ verifying*

$$\tau' \leq \tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + 2q_{\mathfrak{V}} + 2)(T_{exp}(\mathbb{G}_1) + O(1))$$

and a τ'-$\mathcal{PR}_2$-adversary $\mathcal{B}$ against Gen *such that*

$$\mathbf{Succ}^{\mathcal{PR}_2}_{\mathsf{Gen},\mathcal{B}} \geq \frac{\mathbf{Succ}^{\mathsf{UDVS\text{-}EF\text{-}CMA}}_{\mathsf{UDVS\text{-}BLS},\mathcal{A}}}{(1 + 6 \cdot q_{\mathfrak{S}} \cdot 2^{f_r})(q_{\mathfrak{V}} + 1)}.$$

2. *For all $(\tau, q_{\mathfrak{S}}, q_{\mathfrak{V}})$-*UDVS-$\Psi$-CMA*-distinguisher $\mathcal{A}$ against* UDVS-BLS(Gen, f_r, $\mathcal{O}_{\mathfrak{H}}$) *where $\mathcal{O}_{\mathfrak{H}}$ is a $q_{\mathfrak{H}}$-random oracle, there exists $\tau' \in \mathcal{F}(\mathbb{N}, \mathbb{N})$ verifying,*

$$\tau' \leq \tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + 2q_{\mathfrak{V}} + 1)(T_{exp}(\mathbb{G}_1) + O(1)),$$

and a τ'-$\mathcal{PR}_3$-distinguisher $\mathcal{B}$ against Gen *such that*

$$\mathbf{Avan}^{\mathcal{PR}_3}_{\mathsf{Gen},\mathcal{B}} \geq \frac{\mathbf{Avan}^{\mathsf{UDVS\text{-}\Psi\text{-}CMA}}_{\mathsf{UDVS\text{-}BLS},\mathcal{A}}}{2} - \frac{q_{\mathfrak{S}} + q_{\mathfrak{V}}}{2^{f_r}}.$$

Proof. 1. The algorithm $\mathcal{B}$, which takes as input $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi)$ output by $\mathsf{Gen}(k)$, $X \in \mathbb{G}_1$ and $(P_2, Y, Z) \in \mathbb{G}_2^2$ and tries to output $(R_1, R_2) \in \mathbb{G}_1 \times \mathbb{G}_2$ such that $\langle R_1, R_2\rangle = [xyz]\langle P_1, P_2\rangle$, where $P_1 = \psi(P_2)$, $X = [x]P_1$, $Y = [y]P_2$ and $Z = [z]P_2$. Our exact security reduction relies on two clever techniques from [7, 17]:

– Following a well-known technique due to Coron [7], a random coin with expected value $\lambda \in [0, 1]$ decides whether $\mathcal{B}$ introduces the challenge in the answer to the random oracle or an element with a known preimage. For the optimal value of λ, this introduce the (small) loss factor $(1 + 6 \cdot q_{\mathfrak{S}} \cdot 2^{f_r})$ in the success probability.

– Using an approach due to Ogata, Kurosawa and Heng [17], introduced to analyze the security of Chaums undeniable signatures, we do not need a decisional oracle to simulate the verification queries. The idea is that, unless UDVS-BLS is not unforgeable, all verification queries necessarily involve designated verifier signatures that were obtained from signing oracles (and can be readily checked) or that are invalid. $\mathcal{B}$'s strategy is to guess which verification query involves a forged signature and reject signatures involved in all other queries. This is done at the expense of losing the factor $(q_{\mathfrak{V}}+1)$ in $\mathcal{B}$'s probability of success.

For the ease of presentation, let us (at first) assume that $\mathcal{B}$ has an access to a decisional oracle for the problem $\mathcal{PR}_3$ (following Okamoto-Pointcheval's so called gap-problems [18]).

Exp_1 $\mathcal{B}_1$ plays the role of the challenger in the experiment $\mathbf{Exp}^{\mathsf{UDVS\text{-}EF\text{-}CMA}}_{\mathsf{UDVS\text{-}BLS},\mathcal{A}}$ of the definition 2, in the random oracle model. It plugs the bilinear structure underlying its problem instance in the public parameters $\mathcal{P}$. $\mathcal{B}_1$ simulate the random oracle $\mathcal{O}_{\mathfrak{H}}$ by storing the queries made by $\mathcal{A}$ into a list denoted H-List (which contains at most $(q_{\mathfrak{H}}(k) + q_{\mathfrak{S}}(k) + q_{\mathfrak{V}}(k) + 1)$ 4-tuples). $\mathcal{B}_1$ manage a counter c (with initial value 0) and for each signing, verifying or hashing query, $\mathcal{B}_1$ executes the routing Message.

```
Initialization $k$
    $c \leftarrow 0$
    $\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_1, P_2, n_r]$,
    $(u_a, u_b) \xleftarrow{R} [\![1, q-1]\!]^2$
    $(U_a, U_b) \leftarrow ([u_a]P_2, [u_b]P_2)$
    run $\mathcal{A}(\mathcal{P}, U_a, U_b) \rightsquigarrow (m^\star, \tau^\star)$.
Message $m$
    if $\exists i \in [\![1, c]\!]$, $m = m_i$
        then return $i$
        else $c \leftarrow c + 1$
            $m_c \leftarrow m$ ; $\mathcal{L}_c \leftarrow \varepsilon$ ; $\nu \xleftarrow{R} [0, 1]$
            while $\nu < \lambda$ and $\#\mathcal{L}_c \leq q_{\mathfrak{S}}(k)$ do $\rho \xleftarrow{R} \{0,1\}^{n_r}$,
                                                      $\mathcal{L}_c \leftarrow \mathcal{L}_c \cup \{\rho\}$,
                                                      $\nu \xleftarrow{R} [0, 1]$
            return $c$.
```

The oracle queries are then simulated by $\mathcal{B}_1$ in a classical way:

```
Simulation of the oracles
    • $\mathcal{O}_{\mathfrak{H}}(m, r)$: if $\exists R, (m, r, R, ?) \in$ H-List
                     then return $R$
                     else $\alpha \xleftarrow{R} [\![1, q-1]\!]$, $i \leftarrow$ Message$(m)$
                         if $r \in \mathcal{L}_i$ then $R \leftarrow [\alpha]P_1$ else $R \leftarrow [\alpha]X$
                         H-List $\leftarrow$ H-List $\cup \{(m_i, r, R, \alpha)\}$,
                         return $R$
    • $\mathfrak{S}(m)$: $i \leftarrow$ Message$(m)$
            if $\mathcal{L}_i = \emptyset$ then return $\perp$
                     else $r \xleftarrow{R} \mathcal{L}_i$ ; $\mathcal{L}_i \leftarrow \mathcal{L}_i \setminus \{r\}$
                         $\mathcal{O}_{\mathfrak{H}}(m, r)$ ; find $(m_i, r, R, \alpha)$ in H-List
                         $S \leftarrow [\alpha]U_a$
                         return $(r, S)$
    • $\mathfrak{V}(m, \tau)$: $i \leftarrow$ Message$(m)$,
            $(r, Q_1, Q_2) \leftarrow \tau$
            $\mathcal{O}_{\mathfrak{H}}(m, r)$ ; find $(m_i, r, R, \alpha)$ in H-List
            return $\mathcal{PR}_3(R, U_a, U_b, Q_1, Q_2)$.
```

In the random experiment Exp_i, for $i \in \{1, 2\}$, let us denote $\mathsf{F_i}$, the event

$$\text{“}m^\star \notin \mathcal{Q}_{\mathfrak{S}} \text{ and } \mathsf{UDVS\text{-}BLS.DVerify}(\mathcal{P}, m^\star, \mathsf{pk_s}, \tau^\star, \mathsf{pk_c}, \mathsf{sk_c}) = 1.\text{”}$$

Compared to the definition 2, the distribution of $\mathcal{A}$'s entries is unchanged and the simulation of the oracles $\mathcal{O}_{\mathfrak{H}}$ and $\mathfrak{V}$ is perfect. Moreover, $\mathcal{B}_1$ answers without aborting to all signature queries with probability $\lambda^{q_{\mathfrak{S}}(k)}$, and therefore we have

$$\Pr[\mathsf{F}_1] \geq \lambda^{q_{\mathfrak{S}}(k)} \mathbf{Succ}^{\mathsf{UDVS\text{-}EF\text{-}CMA}}_{\mathsf{UDVS\text{-}BLS},\mathcal{A}}(k).$$

Exp_2 $\mathcal{B}_2$ replace in the following the public keys U_a and U_b furnished to $\mathcal{A}$ by the values Y and Z of unknown discrete logarithms (in base P_2).

```
Initialization $k$
    $\mathcal{P} \leftarrow [(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, \langle\cdot,\cdot\rangle, \psi), P_2, n_r]$,
    $(U_a, U_b) \leftarrow (Y, Z)$
    run $\mathcal{A}(\mathcal{P}, U_a, U_b) \rightsquigarrow (m^\star, \tau^\star)$.
```

When $\mathcal{A}$ returns the pair $(m^\star, \tau^\star)$, with $\tau^\star = (r^\star, Q_1^\star, Q_2^\star)$, the algorithm $\mathcal{B}_2$ executes a hash query $\mathcal{O}_{\mathfrak{H}}(m^\star, r^\star)$ and gets $i \in [\![0, c]\!]$ such that $m^\star = m_i$ and $(m^\star, r^\star, R, \alpha)$ in H-List. The algorithm $\mathcal{B}_2$ aborts its execution (by returning $\perp$) if $r^\star \in \mathcal{L}_i$. Otherwise, $\mathcal{B}_2$ returns the pair $([\alpha^{-1}]Q_1^\star, Q_2^\star)$.

```
End of $\mathcal{B} = \mathcal{B}_5$'s execution $(m^\star, \tau^\star)$
    $(r^\star, Q_1^\star, Q_2^\star) \leftarrow \tau^\star$
    $i \leftarrow$ Message$(m)$
    $\mathcal{O}_{\mathfrak{H}}(m, r)$
    find $(m_i, r, R, \alpha)$ in H-List
    if $r^\star \in \mathcal{L}_i$ then return $\perp$
              else return $([\alpha^{-1}]Q_1^\star, Q_2^\star)$.
```

The probability that $r^\star \notin \mathcal{L}_i$ is independent from i and equal to

$$F(\lambda) = [\lambda(1 - 2^{-n_r})]^{q_{\mathfrak{S}}(k)} + (1 - \lambda) \sum_{j=1}^{q_{\mathfrak{S}}(k)-1} [\lambda(1 - 2^{-n_r})]^j.$$

By the simulation, if $r^\star \notin \mathcal{L}_i$ and if the event F_2 holds, we have $R = [\alpha]X$ and if $\tau^\star$ is a valid forgery $\langle Q_1^\star, Q_2^\star\rangle = \langle P_1, P_2\rangle^{xyz\alpha^\star}$. Therefore, the pair $([\alpha^{-1}]Q_1^\star, Q_2^\star)$ is the solution of the problem $\mathcal{PR}_2$ instance.
The security analysis shows that $\mathcal{B} = \mathcal{B}_2$ satisfies

$$\mathbf{Succ}^{\mathcal{PR}_2}_{\mathsf{Gen},\mathcal{B}}(k) \geq \lambda^{q_{\mathfrak{S}}(k)} F(\lambda) \mathbf{Succ}^{\mathsf{UDVS\text{-}EF\text{-}CMA}}_{\mathsf{UDVS\text{-}BLS},\mathcal{A}}(k)$$

and runs in time at most $\tau' \leq \tau + (q_{\mathfrak{H}} + q_{\mathfrak{S}} + q_{\mathfrak{V}} + 1)(T\text{exp}(\mathbb{G}_1) + O(1)) + T\text{exp}(\mathbb{G}_3)$ while making at most $q_{\mathfrak{V}}(k)$ queries to the decisional oracle $\mathcal{PR}_3$. An easy computation gives proves the existence of a value λ_0 such that $\lambda_0^{q_{\mathfrak{S}}(k)} F(\lambda_0) \geq (1 + 6 \cdot q_{\mathfrak{S}} \cdot 2^{f_r})$ (see [7]).
In this reduction, if the decisional oracle for the problem $\mathcal{PR}_3$ returns 1 for a 5-tuple (R, Y, Z, Q_1, Q_2) associated to a verifying query on a pair $(m_i, (r, Q_1, Q_2))$ and if $r \notin \mathcal{L}_i$, then the pair $([\alpha^{-1}]Q_1, Q_2)$ is a solution of the problem $\mathcal{PR}_2$ and there is no need to continue the execution of $\mathcal{A}$. By using this remark, it is possible to prove the resistance to forgery of the scheme to the problem $\mathcal{PR}_2$ without using the decisional oracle.

A verifying query (made by $\mathcal{A}$ or $\mathcal{B}$ at the end of its execution) on a pair $(m_i, (r, Q_1, Q_2))$ is said *special* if r does not belong to the list $\mathcal{L}_i$. Let us denote A the event: "One special verifying query is made in the random experiment Exp_5", $\mathsf{A_i}$ the event "The first special verifying query in the experiment Exp_5 is the i-th", for $i \in [\![1, q_{\mathfrak{V}}(k)]\!]$ and $A_{q_{\mathfrak{V}}(k)+1}$ the event "The first special verifying query in the experiment Exp_5 is the one on the pair $(m^\star, \tau^\star)$". We have

$$\mathsf{A} = \bigsqcup_{i=1}^{q_{\mathfrak{V}}(k)+1} \mathsf{A_i} \text{ et } \widetilde{\mathsf{F}_2} \subseteq \mathsf{A}.$$

where $\widetilde{\mathsf{F}_2}$ is the event "$\mathsf{UDVS\text{-}BLS.DVerify}(\mathcal{P}, m^\star, \mathsf{pk_s}, \tau^\star, \mathsf{pk_c}, \mathsf{sk_c}) = 1$, $m^\star = m_i \notin \mathcal{Q}_{\mathfrak{S}}$ and $r^\star \notin \mathcal{L}_i$."
In this variant, the algorithm $\mathcal{B}$ picks uniformly at random an integer $v \in [\![1, q_{\mathfrak{V}}(k) + 1]\!]$ at the beginning of its execution. For each verifying queries on (m_i, τ) where $\tau = (r, Q_1, Q_2)$, $\mathcal{B}$ gets the 4-tuple (m_i, r, R, α) corresponding to the hash value of (m_i, r) and

– if $r \in \mathcal{L}_i$ then $\mathcal{B}$ returns 1 if and only if $\langle Q_1, Q_2\rangle = \langle \psi(U_a), U_b\rangle^\alpha$;
– if $r \notin \mathcal{L}_i$ and if the verifying query is not the v-th, then $\mathcal{B}$ returns 0;
– if $r \notin \mathcal{L}_i$ and if the verifying query is the v-th, $\mathcal{B}$ stop $\mathcal{A}$'s execution and returns the pair $([\alpha]Q_1, Q_2)$.

If the event A_v occurs then the simulation of the oracles done by $\mathcal{B}$ until the v-th verifying query is indistinguishable from the previous one and $([\alpha]Q_1, Q_2)$ is actually the solution to the instance (P_2, X, Y, Z) of the problem $\mathcal{PR}_2$. Consequently, we have

$$\begin{aligned}\mathbf{Succ}^{\mathcal{PR}_2}_{\mathsf{Gen},\mathcal{B}}(k) \geq \sum_{i=1}^{q_{\mathfrak{V}}(k)+1} \Pr[\mathsf{A_i}] \cdot \Pr[i = v] &= \frac{1}{q_{\mathfrak{V}}(k)+1} \sum_{i=1}^{q_{\mathfrak{V}}(k)+1} \Pr[\mathsf{A_i}] \\ &= \frac{\Pr[\mathsf{A}]}{q_{\mathfrak{V}}(k)+1} \\ &\geq \frac{\Pr[\widetilde{\mathsf{F}_2}]}{q_{\mathfrak{V}}(k)+1}.\end{aligned}$$

This variant of $\mathcal{B}$ does not make any call to the oracle DBDH and its execution time is increased by at most one exponentiation in the group $\mathbb{G}_3$ by each query to the oracle $\mathfrak{V}$. This gives the claimed result.

2. The proof is more or less routine (see [14, 17] for instance) and therefore left to the reader.

Remark 6. If the public verification is desirable in an application (*e.g.* to design a UMDVS scheme) or if the anonymity property is not necessary, the unforgeability of the protocol UDVS-BLS can be reinforced. It is indeed possible to add a fourth element to the signature (namely $Q_3 = [t^{-1}]P_2 \in \mathbb{G}_2$ allowing the public verification) in such a way that the scheme obtained is really close to the protocol UDVS-BB. Since a designated verifier signature for the protocol UDVS-BLS can be readily derived from one for this scheme and since the underlying signature scheme is the same, we get immediately that forging a signature for the latter scheme is at least as hard as for UDVS-BLS. Under the knowledge-of-exponent-assumption in $\mathbb{G}_2$, we can also prove, in the standard security model, the resistance to forgery of the scheme assuming only the EF-CMA-security of the underlying signature scheme BLS.

5 Extension of the schemes to the Multi-verifier setting

At Crypto'03 rump session, Desmedt opened the question to allow several designated verifiers in designated verifier signatures. The first step towards this problem was made in [13] with the introduction of the *multi designated verifiers signature* primitive and some concrete realizations of it. The notion of *universal multi designated verifier signatures* was naturally proposed shortly afterwards in [16].

5.1 UDVS-BB

Let $n \in \mathbb{N}$. The scheme UDVS-BB can be seen as a "discrete-log two-party ring signatures" and therefore, following the generic construction from [13], it can readily be extended into a universal n-designated verifier signature schemes: the algorithm VKeyGen remains unchanged and in the signing algorithm, the verifying public key $\mathsf{pk_v}$ is simply replaced by the sum of the n verifying public keys

$$\mathsf{pk_{v1}} + \cdots + \mathsf{pk_{v\mathit{n}}} = [\mathsf{sk_{v1}} + \cdots + \mathsf{sk_{v\mathit{n}}}]P_2.$$

Using a multi-party computation and the algorithm UDVS-BB.Fake, the designated verifiers can cooperate to produce an n-designated verifier signature from $\mathsf{pk_s}$ to $(\mathsf{pk_{v1}}, \ldots, \mathsf{pk_{v\mathit{n}}})$. This fact, with the source hiding property of UDVS-BB ensure the same property for the multi-user protocol. Finally, since the algorithm UDVS-BB.DVerify is public (*i.e.* does not require the verifying secret key) the algorithm DVerify is identical in the multi-user setting with the verifying public key $\mathsf{pk_v}$ replaced again by the sum of the n verifying public keys $\mathsf{pk_{v1}} + \cdots + \mathsf{pk_{v\mathit{n}}}$. In particular, it is very efficient and does not require interaction between the designated verifiers. It is worth noting, that in order to avoid well-known *rogue key attacks*, the users should prove the knowledge of their secret key (in the registered public key model, for instance).

5.2 UDVS-BLS

The scheme UDVS-BLS is not publicly verifiable and therefore it does not enter in the generic construction proposed in [13]. However, it is possible to adapt this scheme in order to design a universal n-designated verifier signature scheme for all integer $n \geq 1$. With the previous notations, suppose that Alice (*resp.* the n verifiers) has published a public key $U_a = [u_a]P_2$ (*resp.* $U_{b_i} = [u_{b_i}]P_2$ for $i \in [\![1, n]\!]$) and that the pair $\sigma = (r, S) \in [\![1, q-1]\!] \times \mathbb{G}_1$ is a signature BLS produced by Alice on a message m. If Cindy wants to designate σ to the set of the n verifiers, she picks uniformly at random an integer $t \in [\![1, q-1]\!]$ and sets $Q_0 = [t]S$ and for all $i \in [\![1, n]\!]$, $Q_i = [t^{-1}]U_{b_i}$. The $(n+1)$-tuple $\tau = (r, Q_0, Q_1, \ldots, Q_n)$ is the multi-designated verifier signature on m.

The pairing insures the correctness of the scheme since the i-th verifier can check the consistency of the multi-DVS by checking for all $j \in [\![1, n]\!] \setminus \{i\}$ if the equality $\langle \psi(U_j), Q_i \rangle = \langle \psi(U_i), Q_j \rangle$, holds and then ascertain its validity thanks to its knowledge of its secret key by verifying the equality: $\langle Q_0, Q_i \rangle = \langle [u_{b_i}]\mathcal{H}(m, r), U_a \rangle$.

The security properties of the scheme UDVS-BLS(n) are similar to those of the scheme UDVS-BLS. In the security reduction of unforgeability, a factor $1/n$ is lost. This factor corresponds to the bet made by the algorithm $\mathcal{B}$ on the public key that will not corrupt the adversary $\mathcal{A}$. Once this choice has been made, the proof is identical to the one of the theorem 2 and we can easily prove the unforgeability and the anonymity of this scheme assuming the intractability in Gen of the problems $\mathcal{PR}_2$ and $\mathcal{PR}_3$ (respectively).

Acknowledgements

It is a pleasure to acknowledge Fabien Laguillaumie and Benoît Libert for their great comments and simplifying suggestions on a preliminary version of this paper. I am also grateful to Willy Susilo and Rui Zhang for providing a copy of their papers [16, 23]. Finally, I would like to thank the referee for its careful reading of this paper.

References

1. B. Barak, Y. Lindell, and S. Vadhan, *Lower Bounds for Non-Black-Box Zero Knowledge.*, Proceedings of the 44th IEEE Symposium on Foundations of Computer Science (FOCS 2003) (M. Sudan, ed.), IEEE Computer Society, 2003, pp. 384–393.

2. M. Bellare and A. Palacio, *The Knowledge-of-Exponent Assumptions and 3-Round Zero-Knowledge Protocols.*, Advances in Cryptology - Crypto 2004 (M. K. Franklin, ed.), Lect. Notes Comput. Sci., vol. 3152, Springer, 2004, pp. 273–289.
3. M. Bellare and P. Rogaway, *Random Oracles are Practical: A Paradigm for Designing Efficient Protocols.*, Proceedings of the First ACM Conference on Computer and Communications Security (D. Denning, R. Pyle, R. Ganesan, R. Sandhu, and V. Ashby, eds.), ACM Press, 1993, pp. 62–73.
4. D. Boneh and X. Boyen, *Short Signatures Without Random Oracles.*, Advances in Cryptology - Eurocrypt 2004 (C. Cachin and J. Camenisch, eds.), Lect. Notes Comput. Sci., vol. 3027, Springer, 2004, pp. 56–73.
5. D. Boneh, B. Lynn, and H. Shacham, *Short Signatures from the Weil Pairing.*, J. Cryptology **17** (2004), no. 4, 297–319.
6. D. Chaum and H. van Antwerpen, *Undeniable Signatures.*, Advances in Cryptology - Crypto'89 (G. Brassard, ed.), Lect. Notes Comput. Sci., vol. 435, Springer, 1990, pp. 212–216.
7. J.-S. Coron, *Optimal Security Proofs for PSS and Other Signature Schemes.*, Advances in Cryptology - Eurocrypt 2002 (L. R. Knudsen, ed.), Lect. Notes Comput. Sci., vol. 2332, Springer, 2002, pp. 272–287.
8. I. B. Damgård, *Towards Practical Public Key Systems Secure Against Chosen Ciphertext Attacks.*, Advances in Cryptology - Crypto'91 (J. Feigenbaum, ed.), Lect. Notes Comput. Sci., vol. 576, Springer, 1992, pp. 445–456.
9. K. Gjøsteen, *A New Security Proof for Damgård's ElGamal.*, Topics in Cryptology - CT-RSA 2006 (D. Pointcheval, ed.), Lect. Notes Comput. Sci., vol. 3860, Springer, 2006, pp. 150–158.
10. S. Goldwasser, S. Micali, and R. L. Rivest, *A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks.*, SIAM J. Comput. **17** (1988), no. 2, 281–308.
11. M. Jakobsson, K. Sako, and R. Impagliazzo, *Designated Verifier Proofs and Their Applications.*, Advances in Cryptology - Eurocrypt'96 (U. M. Maurer, ed.), Lect. Notes Comput. Sci., vol. 1070, Springer, 1996, pp. 143–154.
12. F. Laguillaumie, B. Libert, and J.-J. Quisquater, *Universal Designated Verifier Signatures Without Random Oracles or Non-Black Box Assumptions.*, Fifth Conference on Security and Cryptography for Networks, SCN 2006, to appear, 2006.
13. F. Laguillaumie and D. Vergnaud, *Multi-designated Verifiers Signatures.*, Information and Communications Security, Sixth International Conference, ICICS 2004 (J. Lopez, S. Qing, and E. Okamoto, eds.), Lect. Notes Comput. Sci., vol. 3269, Springer, 2004, pp. 495–507.
14. ———, *Designated Verifier Signatures: Anonymity and Efficient Construction from* any *Bilinear Map.*, Fourth Conference on Security in Communication Networks, SCN 2004 (C. Blundo and S. Cimato, eds.), Lect. Notes Comput. Sci., vol. 3352, Springer, 2005, pp. 107–121.
15. H. Lipmaa, G. Wang, and F. Bao, *Designated Verifier Signature Schemes: Attacks, New Security Notions and a New Construction.*, Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005 (L. Caires and L. Monteiro, eds.), Lect. Notes Comput. Sci., vol. 3580, Springer, 2005, pp. 459–471.
16. C. Y. Ng, W. Susilo, and Y. Mu, *Universal Designated Multi Verifier Signature Schemes*, International Workshop on Security in Networks and Distributed Systems, SNDS 2005, IEEE Press, 2005, pp. 305–309.
17. W. Ogata, K. Kurosawa, and S.-H. Heng, *The Security of the FDH Variant of Chaum's Undeniable Signature Scheme*, IEEE Trans. Inf. Theory **52** (2006), no. 5, 2006 – 2017.
18. T. Okamoto and D. Pointcheval, *The Gap-Problems: a New Class of Problems for the Security of Cryptographic Schemes.*, 4th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2001 (K. Kim, ed.), Lect. Notes Comput. Sci., vol. 1992, Springer, 2001, pp. 104–118.
19. V. Shoup, *OAEP Reconsidered.*, J. Cryptology **15** (2002), no. 4, 223–249.
20. R. Steinfeld, L. Bull, H. Wang, and J. Pieprzyk, *Universal Designated-Verifier Signatures.*, Advances in Cryptology - Asiacrypt 2003 (C.-S. Laih, ed.), Lect. Notes Comput. Sci., vol. 2894, Springer, 2003, pp. 523–542.
21. D. Vergnaud, *Approximation diophantienne et courbes elliptiques. Protocoles asymétriques d'authentification non-transférable.*, Ph.D. thesis, Université de Caen, november 2006.
22. ———, *New Extensions of Pairing-based Short Signatures into Universal Designated Verifier Signatures.*, 33rd International Colloquium on Automata, Languages and Programming, ICALP 2006 (M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, eds.), Lect. Notes Comput. Sci., vol. 4052, Springer, 2006, pp. 58–69.
23. R. Zhang, J. Furukawa, and H. Imai, *Short Signature and Universal Designated Verifier Signature without Random Oracles.*, Applied Cryptography and Network Security, Third International Conference, ACNS 2005 (J. Ioannidis, A. D. Keromytis, and M. Yung, eds.), Lect. Notes Comput. Sci., vol. 3531, Springer, 2005, pp. 483–498.