Universal Designated-Verifier Signatures

Ron Steinfeld[1], Laurence Bull[2], Huaxiong Wang[1], and Josef Pieprzyk[1]

[1] Dept. of Computing, Macquarie University, North Ryde, NSW 2109, Australia
{rons,hwang,josef}@comp.mq.edu.au
http://www.ics.mq.edu.au/acac/

[2] School of Computer Science and Software Engineering, Monash University, Melbourne, Australia
{laurence.bull@csse.monash.edu.au}

Abstract. Motivated by privacy issues associated with dissemination of signed digital certificates, we define a new type of signature scheme called a 'Universal Designated-Verifier Signature' (UDVS). A UDVS scheme can function as a standard publicly-verifiable digital signature but has additional functionality which allows *any* holder of a signature (not necessarily the signer) to *designate* the signature to any desired *designated-verifier* (using the verifier's public key). Given the designated-signature, the designated-verifier can verify that the message was signed by the signer, but is unable to convince anyone else of this fact.
We propose an efficient *deterministic* UDVS scheme constructed using any bilinear group-pair. Our UDVS scheme functions as a standard Boneh-Lynn-Shacham (BLS) signature when no verifier-designation is performed, and is therefore compatible with the key-generation, signing and verifying algorithms of the BLS scheme. We prove that our UDVS scheme is secure in the sense of our unforgeability and privacy notions for UDVS schemes, under the Bilinear Diffie-Hellman (BDH) assumption for the underlying group-pair, in the random-oracle model. We also demonstrate a general constructive equivalence between a class of unforgeable and unconditionally-private UDVS schemes having unique signatures (which includes the *deterministic* UDVS schemes) and a class of ID-Based Encryption (IBE) schemes which contains the Boneh-Franklin IBE scheme but not the Cocks IBE scheme.

1 Introduction

In the modern world, one can find many examples of *user certification* systems. In these systems, a trusted *Certification Authority* (CA) issues signed certificates to *users*. Typically, the signed certificate attests to the truth of certain statements and attributes linked to the identity of the user to which the certificate is issued. A user Alice can present her certificate to any interested verifier Bob, who can in turn verify the CA's signature and become convinced of the truth of the statements contained in the certificate. Real-life examples include the issuing of birth certificates, driving licences and academic transcripts.

In an electronic world, user certification systems can be implemented through the use of digital signatures. The ease of copying and transmitting electronic certificates in such implementations is of great convenience to users; Alice can simply send a copy of her certificate to any interested verifier Bob. On the other hand, this same ease of distribution applies to Bob as well, who can easily disseminate Alice's certificate and convince an unlimited number of *third-party* verifiers about the truth of the statements concerning Alice contained in the certificate. This possibility poses a serious threat to Alice's *privacy.* Once Alice sends out her certificate to Bob she no longer has any control over the number of entities besides Bob who can not only learn all the statements about Alice contained in the certificate, but also become *convinced* about the truth of these statements by verifying the CA's signature on the certificate.

In this paper, we define a special type of digital signature scheme called a *Universal Designated-Verifier Signature* (UDVS) scheme, which directly addresses the above user privacy issue in user certification systems. Our scheme protects a user's privacy, and yet maintains a similar convenience of use for the user and for the certificate issuer CA as in certification systems using standard digital signatures. In a UDVS scheme, a user Alice is issued a signed certificate by the CA. When Alice wishes to send her certificate to a verifier Bob, she uses Bob's public key to transform the CA's signature into a *designated signature* for Bob, using the UDVS scheme's *designation* algorithm, and sends the certificate along with the *designated-signature* to Bob. Bob can use the CA's public key to verify the designated signature on the certificate, but is unable to use this designated signature to convince any other *third-party* that the certificate was signed by the CA, even if Bob is willing to reveal his secret-key to the *third-party.* This is achieved because Bob's secret-key allows him to forge designated signatures by himself, so the third-party is unable to tell who produced the signature (whereas Bob can, because he knows that he *didn't* produce it). Therefore, through the use of a UDVS scheme, the user Alice's privacy is preserved in the sense that Bob is unable to disseminate *convincing* statements about Alice (Of course, nothing prevents Bob from revealing the certificate statements themselves to any third-party, but the third-party will be unable to tell whether these statements are authentic, i.e. whether they have been signed by the CA or not).

We define quantitative notions of security for both the unforgeability and the privacy provided by UDVS schemes. We then propose an efficient UDVS scheme constructed from any bilinear group-pair, and we prove that this scheme satisfies our security requirements: it achieves perfect unconditional privacy and is unforgeable in the random-oracle model, assuming that the Bilinear Diffie-Hellman (BDH) assumption holds for the underlying bilinear group-pair. Our scheme has the attractive property that its signing, designation, and verification algorithms are all *deterministic.* We also show a general result which establishes a constructive equivalence between a class of unconditionally-private UDVS schemes possessing unique signatures (which contains all deterministic schemes) and a class of strongly-secure Identity-Based Encryption (IBE) schemes which contains the Boneh-Franklin IBE scheme [2], but not the Cocks IBE scheme [14]. Proofs of some statements have been omitted from the appendix due to lack of space.

They can be found in the full version of the paper uploaded to the IACR e-print archive.

1.1 Related Work

Our concept of UDVS schemes can be viewed as an application of the general idea of *designated-verifier proofs*, introduced by Jakobsson, Sako and Impagliazzo [21], where a prover non-interactively designates a proof of a statement to a verifier, in such a way that the verifier can simulate the proof by himself with his secret key and thus cannot transfer the proof to convince anyone else about the truth of the statement, yet the verifier himself is convinced by the proof. The authors of [21] also propose a *designated-verifier non-interactive undeniable signature*, in which the three-move zero-knowledge signature confirmation protocol of an undeniable signature [11] (converted to be non-interactive in the random-oracle model via the Fiat-Shamir heuristic [16]) is modified to be designated-verifier by replacing the commitment with a *trapdoor commitment* [7], in which the verifier's secret key is the trapdoor. However, the resulting scheme in [21] allows designation of signatures only by the *signer* (since designation requires the signer's secret key), whereas our UDVS scheme allows *anyone* who obtains a signature to designate it; this is what we mean by the term *universal* in the name 'Universal Designated-Verifier Signatures'. As we explain in Section 4.1, the idea in [21] of using a trapdoor commitment in a non-interactive zero-knowledge proof can also be used in principle to convert generic digital signature schemes into UDVS schemes. However, the use of a zero-knowledge proof results in a designation algorithm which is randomized, and typically inefficient. In contrast, we show that using bilinear group-pairs one can avoid zero-knowledge proofs and construct a UDVS scheme which has a deterministic and efficient designation algorithm.

There have been other approaches proposed to address the privacy threat associated with dissemination of verifiable signed documents. Chaum and van Antwerpen [13, 11] introduced undeniable signatures for this purpose, which require a signer or confirmer's [12, 27, 25, 9, 17] interactive cooperation to verify a signature, but this approach places significant inconvenience and workload on verifiers and confirmers, compared to an off-line non-interactive verification. There has been substantial work on pseudonym-based *digital credentials* [10, 6, 5, 8] which gives further approaches to enhance user privacy, such as *selective disclosure* of attributes (see also [31]) and *unlinkability* of user transactions. Chameleon signatures [24] allow designation of signatures to verifiers by the *signer*, and in addition allow a signer to prove a forgery by a designated verifier. Ring signatures [28], when restricted to two users, can also be viewed as designated-verifier signatures, where one user is the actual signer and the other user is the designated-verifier who can also forge the two-user ring signature, thus providing the privacy property, called *signer anonymity* in the context of ring signatures. However, signer designation is still performed by the *signer*. Recently, Boneh, Gentry, Lynn and Shacham [3] proposed a ring signature based on bilinear group-pairs and observed that it also allows public conversion of single-signer ring signatures into two-signer ring signatures. Thus, the ring signature scheme in [3] can also be viewed as a UDVS scheme. However, we observe that our proposed UDVS scheme has two advantages over the UDVS scheme in [3]: (1) Our scheme has *deterministic* designation and signing algorithms, and therefore possesses *unique* designated-verifier signatures (unlike the randomized designation scheme in [3]). As we show in Sec. 5, secure UDVS schemes with unique signatures are as hard to construct as secure ID-Based Encryption (IBE) schemes (our scheme is related to the Boneh-Franklin IBE [2]), whereas this is not the case for randomized UDVS schemes, which can be constructed using other methods, (2) Our scheme extends the standard BLS signature [4], whereas the scheme in [3] is built on a modified BLS scheme. Our scheme also has an efficiency advantage in verification compared to [3] (see Section 6.1).

2 Preliminaries

2.1 Notation

We say that a function $f : \mathbb{N} \to \mathbb{R}$ is a *negligible* function if, for any $c > 0$, there exists $k_0 \in \mathbb{N}$ such that $f(k) < 1/k^c$ for all $k > k_0$. We say that a probability function $p : \mathbb{N} \to \mathbb{R}$ is *overwhelming* if the function $q : \mathbb{N} \to \mathbb{R}$ defined by $q(k) = 1 - p(k)$ is a negligible function. For various algorithms discussed, we will define a sequence of integers to measure the *resource parameters* of these algorithms (e.g. running-time plus program length, number of oracle queries to various oracles). All these resource parameters can in general be functions of a *security parameter* k of the scheme. We say that an algorithm A with resource parameters $RP = (r_1, \ldots, r_n)$ is *efficient* if each resource parameter $r_i(k)$ of A is bounded by a polynomial function of the security parameter k, i.e. there exists a $k_0 > 0$ and $c > 0$ such that $r_i(k) < k^c$ for all $k > k_0$. For a probabilistic algorithm A, we use $\mathsf{A}(x; r)$ to denote the output of A on input x with a randomness input r. If we do not specify r explicitly we do so with the understanding that r is chosen statistically independent of all other variables. We denote by $\{\mathsf{A}(x)\}$ the set of outputs of A on input x as we sweep the randomness input for A through all possible strings.

2.2 Bilinear Group-Pairs

Our signature scheme proposed in Section 4.2 is built using a powerful cryptographic tool called a *Bilinear Group-Pair*. In this section we review the definition of a bilinear group-pair, following the definitions of [3]. We refer the reader to [22, 23, 2, 4] for a discussion of how to build a concrete instance of such a group-pair using supersingular elliptic curves, and to [1] for efficient algorithms for computing the bilinear map over these group-pairs.

Definition 1 (Bilinear Group-Pair). *Let (G_1, G_2) denote a pair of groups of prime order $|G_1| = |G_2|$. We call the group-pair (G_1, G_2) a* Bilinear Group-Pair *if the pair (G_1, G_2) has the following properties:*

(1) Efficient Group Operations: *The group operations in* G_1 *and* G_2 *are efficiently computable (in some representation).*
(2) Existence of Efficient Bilinear Map: *There exists an efficiently computable* bilinear map $e : G_1 \times G_2 \to G_T$ *(for some* image group G_T *of order* $|G_T| = |G_1| = |G_2|$*) having the following properties:*
   (a) Bilinearity: $e(u_1^{a_1}, u_2^{a_2}) = e(u_1, u_2)^{a_1 \cdot a_2}$ *for all* $(u_1, u_2) \in G_1 \times G_2$ *and* $(a_1, a_2) \in \mathbb{Z}^2$.
   (b) Non-Degeneracy: $e(u_1, u_2) \neq 1$ *for all* $(u_1, u_2) \in G_1/\{1\} \times G_2/\{1\}$ *(Here* 1 *denotes the identity element in the respective group).*
(3) Existence of Efficient Isomorphism: *There exists an efficiently computable* group isomorphism $\psi : G_1 \to G_2$ *from* G_1 *to* G_2.

Our signature scheme's security relies on the computational hardness of the *Bilinear Diffie-Hellman* (BDH) problem associated with the bilinear group-pair used to construct the scheme. We review the BDH problem, and remark that the Boneh-Franklin ID-Based Encryption scheme [2] and Joux's tripartite key exchange protocol [22] also rely on the hardness of BDH.

Definition 2 (Bilinear Diffie-Hellman (BDH) Problem). *Let* GC *denote a randomized* bilinear group-pair instance generation algorithm*, which on input a security parameter* k*, outputs* (D_G, g_1)*, where* $D_G \in \{0,1\}^*$ *is a description string for a bilinear group-pair* (G_1, G_2)*. We say that the BDH problem is* hard *in group-pairs generated by* GC *if, for any efficient attacker* A*, the probability* $\mathbf{Succ}_{\mathsf{A},\mathsf{BDH}}(k)$ *that* A *succeeds to compute* $K = e(g_1, g_2)^{a \cdot b \cdot c}$ *given* $(D_G, g_1, g_1^a, g_1^b, g_2^c)$ *for uniformly random* $a, b, c \in \mathbb{Z}_{|G_1|}$*, where* $g_2 = \psi(g_1)$*, is a negligible function of* k *(the probability is over* A*'s random coins and the input to* A*). We quantify the insecurity of* BDH *against arbitrary attackers with running-time plus program length* t *by the probability* $\mathbf{InSec}_{\mathsf{BDH}}(t) \stackrel{\mathrm{def}}{=} \max_{\mathsf{A} \in AS_{RP}} \mathbf{Succ}_{\mathsf{A},\mathsf{BDH}}(k)$*, where the set* AS_{RP} *contains all attackers with run-time* t*.*

3 Universal Designated-Verifier Signature (UDVS) Schemes

3.1 Precise Definition of a UDVS Scheme

A Universal Designated Verifier Signature (UDVS) scheme DVS consists of seven algorithms and a 'Verifier Key-Registration Protocol' P_{KR}. All these algorithms may be randomized.

1. **Common Parameter Generation** GC — on input a security parameter k, outputs a string consisting of common scheme parameters cp (publicly shared by all users).
2. **Signer Key Generation** GKS — on input a common parameter string cp, outputs a secret/public key-pair (sk_1, pk_1) for *signer*.

3. **Verifier Key Generation** GKV — on input a common parameter string cp, outputs a secret/public key-pair (sk_3, pk_3) for *verifier*.
4. **Signing** S — on input signing secret key sk_1, message m, outputs *signer*'s publicly-verifiable (PV) signature σ.
5. **Public Verification** V — on input *signer*'s public key pk_1 and message/PV-signature pair (m, σ), outputs verification decision $d \in \{Acc, Rej\}$.
6. **Designation** CDV — on input a *signer*'s public key pk_1, a *verifier*'s public key pk_3 and a message/PV-signature pair (m, σ), outputs a designated-verifier (DV) signature $\widehat{\sigma}$.
7. **Designated Verification** VDV — on input a *signer*'s public key pk_1, *verifier*'s secret key sk_3, and message/DV-signature pair $(m, \widehat{\sigma})$, outputs verification decision $d \in \{Acc, Rej\}$.
8. **Verifier Key-Registration** $\mathsf{P_{KR}} = (\mathsf{KRA}, \mathsf{VER})$ — a protocol between a 'Key Registration Authority' (KRA) and a 'Verifier' (VER) who wishes to register a verifier's public key. On common input cp, the algorithms KRA and VER interact by sending messages alternately from one to another. At the end of the protocol, KRA outputs a pair $(pk_3, Auth)$, where pk_3 is a verifier's public-key, and $Auth \in \{Acc, Rej\}$ is a key-registration authorization decision. We write $P_{KR}(\mathsf{KRA}, \mathsf{VER}) = (pk_3, Auth)$ to denote this protocol's output.

Verifier Key-Reg. Protocol. The purpose of the 'Verifier Key-Registration' protocol is to force the verifier to 'know' the secret-key corresponding to his public-key, in order to enforce the non-transferability privacy property. In this paper we assume the *direct* key reg. protocol, in which the verifier simply reveals his key-pair (sk, pk), and the KRA authorizes it only if $(sk, pk) \in \{\mathsf{GKV}(cp)\}$.[3]

Consistent UDVS Schemes. We require two obvious consistency properties from UDVS schemes. The 'PV-Consistency' property requires that the PV-signatures produced by the *signer* are accepted as valid by the PV-verification algorithm V. The 'DV-Consistency' property requires that the DV-signatures produced by the *designator* using the designation algorithm CDV are accepted as valid by the DV-verification algorithm VDV. We say that a UDVS scheme is *consistent* if it has both of the above consistency properties.

DVSig-Unique UDVS schemes. In this paper we are mainly interested in *DVSig-Unique* UDVS schemes, in which the DV signature $\sigma^*_{dv} = \mathsf{CDV}(pk_1, pk_3, \mathsf{S}(sk_1, m^*))$ on a message m^* by a signer with public key pk_1 (and secret key sk_1) to a verifier with public key pk_3, is uniquely determined by (m^*, pk_1, pk_3).

3.2 Security Properties of UDVS Schemes

3.2.1 Unforgeability In the case of a UDVS scheme there are actually two types of unforgeability properties to consider. The first property, called

[3] The KRA can always perform this check efficiently, since we can assume that the secret key sk contains the randomness input to GKV used to generate it.

called 'PV-Unforgeability', is just the usual existential unforgeability notion under chosen-message attack [20] for the standard PV signature scheme $D = (\mathsf{GC}, \mathsf{GKS}, \mathsf{S}, \mathsf{V})$ induced by the UDVS scheme (this prevents attacks to fool the *designator*). The second property, called 'DV-Unforgeability', requires that it is difficult for an attacker to forge a DV-signature $\widehat{\sigma}^*$ by the signer on a 'new' message m^*, such that the pair $(m^*, \widehat{\sigma}^*)$ passes the DV-verification test with respect to a given designated-verifier's public key pk_3 (this prevents attacks to fool the designated verifier, possibly mounted by a dishonest designator). It is easy to see that, due to the existence of the efficient public-designation algorithm CDV, the 'DV-unforgeability' property implies the 'PV-unforgeability' property[4], although the converse need not hold in general. Indeed, we will see that our proposed UDVS scheme's 'PV-unforgeability' can be proven with a weaker assumption than that needed to prove the 'DV-unforgeability'.

Definition 3 (DV-Unforgeability). *Let* $\mathsf{DVS} = (\mathsf{GC}, \mathsf{GKS}, \mathsf{GKV}, \mathsf{S}, \mathsf{V}, \mathsf{CDV}, \mathsf{VDV}, \mathsf{P}_{\mathsf{KR}})$ *be a UDVS scheme. Let* A *denote a forger attacking the unforgeability of* DVS*. The DV-Unforgeability notion UF-DV for this scheme is defined as follows:*

1. ***Attacker Input:*** *Signer and Verifier's public-keys* (pk_1, pk_3) *(from* $\mathsf{GKS}(k)$,$\mathsf{GKV}(k)$*).*
2. ***Attacker Resources:*** *Run-time plus program-length at most* t*, Oracle access to signer's signing oracle* $\mathsf{S}(sk_1, .)$ *(*q_s *queries), and, if scheme* DVS *makes use of* n *random oracles* $RO_1, \ldots, RO_n$*, allow* q_{RO_i} *queries to the* i*th oracle* RO_i *for* $i = 1, \ldots, n$*. We write attacker's Resource Parameters (RPs) as* $RP = (t, q_s, q_{RO_1}, \ldots, q_{RO_n})$*.*
3. ***Attacker Goal:*** *Output a forgery message/DV-signature pair* $(m^*, \widehat{\sigma}^*)$ *such that:*
   (1) The forgery is valid, i.e. $\mathsf{VDV}(pk_1, sk_3, m^*, \widehat{\sigma}^*) = Acc$*.*
   (2) Message m^* *is 'new', i.e. has not been queried by attacker to* S*.*
4. ***Security Notion Definition:*** *Scheme is said to be* unforgeable *in the sense of UF-DV if, for any efficient attacker* A*, the probability* $\mathbf{Succ}^{UF\text{-}DV}_{\mathsf{A},\mathsf{DVS}}(k)$ *that* A *succeeds in achieving above goal is a negligible function of* k*. We quantify the insecurity of* DVS *in the sense of UF-DV against arbitrary attackers with resource parameters* $RP = (t, q_s, q_{RO_1}, \ldots, q_{RO_n})$ *by the probability*

$$\mathbf{InSec}^{UF\text{-}DV}_{\mathsf{DVS}}(t, q_s, q_{RO_1}, \ldots, q_{RO_n}) \stackrel{\text{def}}{=} \max_{\mathsf{A} \in AS_{RP}} \mathbf{Succ}^{UF\text{-}DV}_{\mathsf{A},\mathsf{DVS}}(k),$$

   where the set AS_{RP} *contains all attackers with resource parameters* RP*.*

[4] Actually, this assumes that $\mathsf{V}(pk_1, m, \sigma) = Acc$ implies $\sigma \in \{\mathsf{S}(sk_1, m)\}$ for all (m, σ). But even if this does not hold, we can always redefine V to verify (m, σ) using pk_1 as follows: compute random key-pair $(sk_3, pk_3) = \mathsf{GKV}(cp)$, compute $\widehat{\sigma} = \mathsf{CDV}(pk_1, pk_3, m, \sigma)$ and return $\mathsf{VDV}(pk_1, sk_3, m, \widehat{\sigma})$. It is easy to see that using this V, DV-Unforgeability implies PV-Unforgeability.

3.2.2 Non-Transferability Privacy Informally, the purpose of the privacy property for a UDVS scheme is to prevent a designated-verifier from using the DV signature σ_{dv} on a message m to produce evidence which convinces a third-party that the message m was signed by the signer. Our model's goal is to capture a setting in which signature holder provides many designated-signatures on m, designated to many verifier public keys of the attacker's choice. We quantify this property using the following privacy attack model. In our model, the attacker is a pair of interacting algorithms $(\mathsf{A}_1, \mathsf{A}_2)$ representing the designated-verifier (DV) and Third-Party (TP), respectively, which run in two stages. At the end of Stage 1, A_1 decides on a message m^* to be signed by the signer. In Stage 2, A_1 obtains up to q_{d1} DV signatures $(\sigma_1, \ldots, \sigma_{q_{d1}})$ by the signer on m^* from a designator oracle, designated to public-keys of A_1's choice (these keys must first be registered by A_1 via key-reg. interactions with the KRA), and tries to use the σ_i's to convince A_2 that the signer signed m^*. At the end of Stage 2, A_2 outputs an estimate $d \in \{\mathsf{yes}, \mathsf{no}\}$ for the answer to the question 'did the signer sign m^*'?

We associate with $(\mathsf{A}_1, \mathsf{A}_2)$ a *convincing measure* $C_{\widehat{\mathsf{A}_1}}(\mathsf{A}_1, \mathsf{A}_2)$ with respect to a forgery strategy $\widehat{\mathsf{A}_1}$, to measure the 'degree' to which A_2 can be convinced by A_1 that the signer signed m^*. It is defined as the *distinguisher advantage* of A_2's estimate d to correctly distinguish between (1) The game yes, where the signer did sign m^* and A_1 obtained one or more DV signatures on m^* from the designator oracle or (2) The game no, where the signer did not sign m^* and A_1 was actually replaced by an efficient forging strategy, called $\widehat{\mathsf{A}_1}$ (which accepts the program for A_1 as input), which aims to "trick" A_2 into believing that the signer signed m^*, without the need to obtain any DV signatures on m^* from the designator oracle. The scheme is said to achieve the privacy property if there is an efficient forgery strategy $\widehat{\mathsf{A}_1}$ such that $C_{\widehat{\mathsf{A}_1}}(\mathsf{A}_1, \mathsf{A}_2)$ is negligible for any efficient attacker pair $(\mathsf{A}_1, \mathsf{A}_2)$.

Definition 4 (PR-Privacy). *Let* $\mathsf{DVS} = (\mathsf{GC}, \mathsf{GKS}, \mathsf{GKV}, \mathsf{S}, \mathsf{V}, \mathsf{CDV}, \mathsf{VDV}, \mathsf{P_{KR}})$ *be a UDVS scheme. Let* $(\mathsf{A}_1, \mathsf{A}_2)$ *denote an attack pair against the privacy of* DVS*. Let* $\widehat{\mathsf{A}_1}$ *denote a forgery strategy. The privacy notion PR for this scheme is defined as follows:*

1. ***Attacker Input:*** *Signer public-key* pk_1 *(where* $(sk_1, pk_1) = \mathsf{GKS}(cp)$*, and* $cp = \mathsf{GC}(k)$*). Note that* $\widehat{\mathsf{A}_1}$ *also accepts the program for* A_1 *as input.*
2. ***Resources for*** $(\mathsf{A}_1, \widehat{\mathsf{A}_1})$***:*** *Run-time* $(t_1, \widehat{t_1})$*, access to signing oracle* $\mathsf{S}(sk_1, .)$ *(up to* $(q_s, \widehat{q_s})$ *queried messages different from* m^**), access to key-reg. protocol interactions with the* KRA *(up to* $(q_k, \widehat{q_k})$ *interactions), access to* A_2 *oracle (up to* $(q_c, \widehat{q_c})$ *messages). In stage 2,* A_1 *also has access to designation oracle* $\mathsf{CDV}(pk_1, ., m^*, \sigma^*)$ *(up to* q_d *queried keys successfully registered with* KRA*), where* $\sigma^* = \mathsf{S}(sk_1, m^*)$ *is a signer's signature on the challenge message* m^* *output by* A_1 *at end of Stage 1. Note that* $\widehat{\mathsf{A}_1}$ *cannot make any designation queries.*
3. ***Resources for*** A_2***:*** *Run-time* t_2*.*

4. ***Attacker Goal:*** *Let* $P(\mathsf{A}_1, \mathsf{A}_2)$ *and* $P(\widehat{\mathsf{A}_1}, \mathsf{A}_2)$ *denote the probabilities that* A_2 *outputs* yes *when interacting with* A_1 *(game* yes*) and* $\widehat{\mathsf{A}_1}$ *(game* no*), respectively. The goal of* $(\mathsf{A}_1, \mathsf{A}_2)$ *is to achieve a non-negligible convincing measure* $C_{\widehat{\mathsf{A}_1}}(\mathsf{A}_1, \mathsf{A}_2) \stackrel{\text{def}}{=} |P(\mathsf{A}_1, \mathsf{A}_2) - P(\widehat{\mathsf{A}_1}, \mathsf{A}_2)|$.
5. ***Security Notion Definition:*** *Scheme is said to achieve* privacy *in the sense of PR if there exists an efficient forgery strategy* $\widehat{\mathsf{A}_1}$ *such that the convincing measure* $C_{\widehat{\mathsf{A}_1}}(\mathsf{A}_1, \mathsf{A}_2)$ *achieved by any efficient attacker pair* $(\mathsf{A}_1, \mathsf{A}_2)$ *is negligible in the security parameter* k*. We quantify the insecurity of* DVS *in the sense of PR against arbitrary attacker pairs* $(\mathsf{A}_1, \mathsf{A}_2)$ *with resources* (RP_1, RP_2) *(attacker set* AS_{RP_1,RP_2}*), with respect to arbitrary forgery strategies* $\widehat{A_1}$ *with resources* $\widehat{RP_1}$ *(attacker set* $AS_{\widehat{RP_1}}$*) by the probability*

$$\mathbf{InSec}^{PR}_{\mathsf{DVS}}(RP_1, \widehat{RP_1}, RP_2) \stackrel{\text{def}}{=} \min_{\widehat{\mathsf{A}_1} \in AS_{\widehat{RP_1}}} \max_{(\mathsf{A}_1,\mathsf{A}_2) \in AS_{RP_1,RP_2}} C_{\widehat{\mathsf{A}_1}}(\mathsf{A}_1, \mathsf{A}_2).$$

If $\mathbf{InSec}^{PR}_{\mathsf{DVS}}(RP_1, \widehat{RP_1}, RP_2) = 0$ *holds for any computationally unbounded* A_2*, it is said to be* perfect unconditional privacy*. If privacy holds when* $\widehat{q_{s1}} = q_{s1}$ *it is said to be* complete *privacy.*

Remark. The above privacy notion handles general UDVS schemes. For more specific schemes, the definition can be simplified. For instance, for schemes using the *direct* key-reg. protocol which have unique signatures, the complete unconditional privacy is equivalent to the existence of an efficient universal forgery algorithm for DV signatures using the verifier's secret key (this is the case for our proposed scheme in this paper, but see Sec. 6.2 for other possibilities).

Lemma 1. *Let* $\mathsf{DVS} = (\mathsf{GC}, \mathsf{GKS}, \mathsf{GKV}, \mathsf{S}, \mathsf{V}, \mathsf{CDV}, \mathsf{VDV}, \mathsf{P}_{\mathsf{KR}})$ *be a UDVS scheme which is DVSig-Unique, and where* P_{KR} *is the direct key-reg. protocol. Then* DVS *achieves complete and perfect unconditional privacy if and only if there exists an efficient universal DV-sig. forgery algorithm* F*, which on any input* $(cp, pk_1, sk_3, pk_3, m^*)$ *(where* $(sk_1, pk_1) \in \{\mathsf{GKS}(cp)\}$ *and* $(sk_3, pk_3) \in \{\mathsf{GKV}(cp)\}$*) computes the unique DV-sig.* $\sigma^*_{dv} = \mathsf{CDV}(cp, pk_1, pk_3, \mathsf{S}(sk_1, m^*))$ *with probability 1.*

4 Proposed UDVS Scheme

4.1 An Inefficient Generic Approach for Constructing UDVS Schemes

Before we present our efficient UDVS scheme, we sketch, as a plausibility argument, the details of a generic (but inefficient) approach for constructing UDVS schemes, based on zero-knowledge designated-verifier proofs of membership [21]. We do not attempt to give a precise definition and proof of security properties for this generic scheme, but we believe this can be done along the outline we sketch below.

The generic construction works as follows. We make use of a standard digital signature scheme $\mathsf{DS} = (\mathsf{GK_S}, \mathsf{S}, \mathsf{V})$ which is secure in the standard CMA sense of existential unforgeability under chosen message attack [20]. We also need a public-key encryption scheme $\mathsf{PKE} = (\mathsf{GK_E}, \mathsf{E}, \mathsf{D})$ which is semantically secure under chosen-plaintext attack(IND-CPA) [19], and a trapdoor commitment scheme TC [7]. The common parameter generation algorithm GC for the UDVS scheme generates an encryption key-pair $(sk_E, pk_E) = \mathsf{GK_E}(k)$ and outputs pk_E as the common parameter. The signer key-generation/signing/PV-verification algorithms for the UDVS scheme are those of the signature scheme DS. The verifier's key-generation algorithm is that of the trapdoor commitment scheme TC. The designation algorithm CDV takes an input common parameter pk_E, message m and its signature σ_{pv}, signer's public key pk_1, and verifier's public key pk_3. The designated signature σ_{dv} is the pair (c, P), where $c = \mathsf{E}(pk_E, \sigma_{pv}; r)$ is the encryption of σ_{pv} under the common public key pk_E (using a random string r), and P is a designated-verifier non-interactive proof that c is in the NP language

$$L_{pk_1,pk_E,m} = \{c : \exists(\sigma, r) \text{ such that } c = \mathsf{E}(pk_E, \sigma; r) \text{ and } \mathsf{V}(pk_1, m, \sigma) = Acc\},$$

consisting of all possible ciphertexts of valid signatures by the signer on the message m. Note that the designator has a witness (σ, r) for membership of c in $L_{pk_1,pk_E,m}$, and hence can use a generic zero-knowledge commit-challenge-response proof of membership for NP languages [18] to prove that $c \in L_{pk_1,pk_E,m}$. By applying the Fiat-Shamir heuristic (replacing the challenge by a hash of the commitments) to make the proof non-interactive and using the verifier's trapdoor commitment in the commit step, the designator can compute the desired designated-verifier proof P. The DV verification algorithm consists of verifying the proof P for the ciphertext c. The DV-unforgeability of the scheme follows (in the random oracle model) from the soundness of the proof and the unforgeability of the underlying standard signature scheme: any forged ciphertext with a valid proof is by soundness a ciphertext of a valid signature and can be decrypted with sk_E to give a forgery for the underlying signature scheme. The (computational) privacy follows from the forgeability of the proof P by the designated-verifier using his secret-key, namely the commitment trapdoor (even for ciphertexts c not in the language $L_{pk_1,pk_E,m}$), and the (computational) simulatability of the ciphertext c by a random string, due to the semantic security of the encryption scheme.

Implementing the above scheme using a generic zero-knowledge NP proof system [18] would yield a very inefficient and randomized designation and inefficient DV verification. For specific choices of the underlying signature and encryption scheme, one may be able to give a more efficient zero-knowledge proof for the language $L_{pk_1,pk_E,m}$ and improve this efficiency to some extent. However, our bilinear scheme below shows how to eliminate zero-knowledge proofs altogether and obtain efficient, deterministic designation.

4.2 An Efficient UDVS Scheme DVSBM Based on Bilinear Group-Pairs

Our proposed UDVS scheme DVSBM based on bilinear group-pairs is given below. It makes use of a cryptographic hash function $H : \{0,1\}^{\leq \ell} \rightarrow G_2$, modelled as a random-oracle. Here ℓ denotes a bound on the message bit-length and $\{0,1\}^{\leq \ell}$ denotes the message space of all strings of length at most ℓ bits. Note that for the basic version of DVSBM we propose the *direct* key registration protocol (see Section 3.1).

1. **Common Parameter Generation** GC. Choose a bilinear group-pair (G_1, G_2) of prime order $|G_1| = |G_2|$ with description string D_G specifying a blinear map $e : G_1 \times G_2 \rightarrow G_T$, isomorphism $\psi : G_1 \rightarrow G_2$ and generators $g_1 \in G_1$ and $g_2 = \psi(g_1) \in G_2$. The common parameters are $cp = (D_G, g_1)$.
2. **Signer Key Generation** GKS. Given cp, pick random $x_1 \in \mathbb{Z}_{|G_1|}$ compute $y_1 = g_1^{x_1}$. The public key is $pk_1 = (cp, y_1)$. The secret key is $sk_1 = (cp, x_1)$.
3. **Verifier Key Generation** GKV. Given cp, pick random $x_3 \in \mathbb{Z}_{|G_1|}$ compute $y_3 = g_1^{x_3}$. The public key is $pk_3 = (cp, y_3)$. The secret key is $sk_3 = (cp, x_3)$.
4. **Signing** S. Given the signer's secret key (cp, x_1), and message m, compute $\sigma = h^{x_1} \in G_2$, where $h = H(m)$. The PV signature is σ.
5. **Public Verification** V. Given the signer's public key (cp, y_1) and a message/PV-sig. pair (m, σ), accept if and only if $e(g_1, \sigma) = e(y_1, h)$, where $h = H(m)$.
6. **Designation** CDV. Given the signer's public key (cp, y_1), a verifier's public key (cp, y_3) and a message/PV-signature pair (m, σ), compute $\widehat{\sigma} = e(y_3, \sigma)$. The DV signature is $\widehat{\sigma}$.
7. **Designated Verification** VDV. Given a signer's public key (cp, y_1), a verifier's secret key (cp, x_3), and message/DV-sig. pair $(m, \widehat{\sigma})$, accept if and only if $\widehat{\sigma} = e(y_1^{x_3}, h)$, where $h = H(m)$.

Consistency. We first demonstrate the consistency of scheme DVSBM. To show the PV-Consistency property, we note that if $\sigma_{pv} \stackrel{\text{def}}{=} \mathsf{S}(sk_1, m) = h^{x_1}$, where $h = H(m)$, then

$$e(g_1, \sigma) = e(g_1, h^{x_1}) = e(g_1, h)^{x_1} = e(g_1^{x_1}, h) = e(y_1, h) \tag{1}$$

by bilinearity, so $\mathsf{V}(pk_1, m, \sigma_{pv}) = Acc$, as required. To show the DV-Consistency property, we note that if $\sigma_{pv} \stackrel{\text{def}}{=} \mathsf{S}(sk_1, m) = h^{x_1}$, where $h = H(m)$, then $\sigma_{dv} \stackrel{\text{def}}{=} \mathsf{CDV}(pk_1, pk_3, m, \sigma_{pv}) = e(y_3, \sigma_{pv})$. Consequently:

$$\widehat{\sigma}_{dv} \stackrel{\text{def}}{=} e(y_1^{x_3}, h) = e(y_3^{x_1}, h) = e(y_3, h)^{x_1} = e(y_3, h^{x_1}) = e(y_3, \sigma_{pv}) = \sigma_{dv} \tag{2}$$

by bilinearity, so $\mathsf{VDV}(pk_1, (sk_3, pk_3), m, \sigma_{dv}) = Acc$, as required. Therefore the scheme DVSBM is *consistent.*

Unforgeability. In the random-oracle model for $H(.)$, we can prove the DV-unforgeability of the scheme assuming the Bilinear Diffie-Hellman (BDH) assumption. The reduction is efficient (no q_H multiplicative cost in insecurity bound) thanks to the random self-reducibility of BDH, by adapting Coron's technique [15] which was originally applied to prove the unforgeability of the $\mathsf{FDH-RSA}$ signature scheme assuming the RSA assumption. We note that the PV-unforgeability of our scheme reduces to the unforgeability of the BLS scheme [4], which was proven in [4] under a weaker assumption than hardness of BDH, namely hardness of the 'co-CDH' assumption.

Theorem 1 (DV-unforgeability of DVSBM). *If the Bilinear Diffie-Hellman problem is hard in the bilinear group-pairs (G_1, G_2) generated by the common-parameter algorithm GC, then the scheme DVSBM is DV-unforgeable (UF-DV notion) in the random-oracle model for $H(.)$. Concretely, the following insecurity bound holds:*

$$\mathbf{InSec}^{UF\text{-}DV}_{\mathsf{DVSBM}}(t, q_s, q_H) \leq \exp(1) \cdot (q_s + 1) \cdot \mathbf{InSec}_{\mathsf{BDH}}(t[B]), \tag{3}$$

where $t[B] = t + (q + q_s + 1) \cdot O(\ell \cdot \log_2 q + T_g \cdot \log_2 |G_1|) + T_\psi + T_e$. Here we define $q \overset{\text{def}}{=} q_H + q_s + 1$ and denote by T_e, T_g, and T_ψ the running time bounds for evaluating the bilinear map e, performing a single group operation in G_1 or G_2, and evaluating the isomorphism ψ, respectively, and we use $\exp : \mathbb{R} \to \mathbb{R}$ to denote the natural exponential function.

Privacy. The privacy achieved by scheme DVSBM is perfect unconditional, because the verifier can easily forge the DV-signatures he is receiving from the designator (as long as the verifier knows his secret-key, which is ensured by the key-registration protocol).

Theorem 2 (Privacy of DVSBM). *The scheme DVSBM achieves complete and perfect unconditional privacy (in the sense of the PR notion). Concretely:*

$$\mathbf{InSec}^{PR}_{\mathsf{DVSBM}}(RP_1, \widehat{RP}_1, \infty) = 0, \tag{4}$$

where $RP_1 = (t_1, q_s, q_k, q_d)$ denotes A_1's resource parameters and $\widehat{RP}_1 = (\widehat{t}_1, \widehat{q}_s, \widehat{q}_k)$ denotes the forgery strategy $\widehat{\mathsf{A}_1}$'s resources, which are given by: $\widehat{t}_1 = t_1 + q_d \cdot O(T_e + T_g \log_2 |G_1| + q_k)$, $\widehat{q}_s = q_s$ (complete privacy), $\widehat{q}_d = q_d$, $\widehat{q}_c = q_c$.

5 General Relationship Between UDVS and ID-Based Encryption Schemes

Readers who are familiar with the Boneh-Franklin ID-Based Encryption scheme [2] may notice an intimate relationship between that scheme and our proposed UDVS scheme DVSBM. In this section we show that this relationship is one instance of a general equivalence between certain subclass of secure UDVS schemes and a certain subclass of secure ID-Based Encryption schemes.

ID-Based Key Encapsulation Mechanism (ID-KEM) Schemes. We review the definition of ID-based encryption schemes (IBE) [2]. Actually, we will formulate our result in terms of a primitive called 'ID-Based Key Encapsulation Mechanism' (ID-KEM), defined analogously to the definition of standard non-ID-based KEMs [30]. An ID-Based Key Encapsulation Mechanism (ID-KEM) consists of 4 algorithms: Setup, $\mathsf{Extract}$, $\mathsf{Encrypt}$, $\mathsf{Decrypt}$: Setup takes as input security parameter k, and returns a system parameter string cp and a master key mk. This is run initially by the 'Private Key Generator' (PKG). $\mathsf{Extract}$ takes as input system parameters cp, master key mk, and a user identity string $ID \in S_{ID}$ and returns a user secret key $sk_{ID} = \mathsf{Extract}(cp, mk, ID)$ corresponding to identity ID. $\mathsf{Encrypt}$ is a randomized algorithm which takes as input system parameters cp, a recipient identity string ID and a random input $r \in S_R$ and returns a pair $(K, c) = \mathsf{Encrypt}(cp, ID; r)$, where $K = \mathsf{Enc_K}(cp, ID; r)$ is an 'session key' (which can be used with a symmetric encryption scheme to encrypt a message) and $c = \mathsf{Enc_c}(cp, ID; r)$ is a ciphertext for K (we call $\mathsf{Enc_K}$ and $\mathsf{Enc_c}$ the *key-computation* and *key-encapsulation* functions induced by $\mathsf{Encrypt}$). Given cp, sk_{ID} and c, $\mathsf{Decrypt}(cp, sk_{ID}, c)$ recovers a session key K. An ID-KEM is *consistent* if $\mathsf{Decrypt}(cp, sk_{ID}, \mathsf{Enc_c}(cp, ID; r)) = \mathsf{Enc_K}(cp, ID; r)$ holds, where $sk_{ID} = \mathsf{Extract}(cp, mk, ID)$, for all (ID, r) and (cp, mk) generated by Setup.

Ephemeral-Key (EK) and Separable ID-KEM Schemes. For constructing UDVS schemes, we need an ID-KEM scheme which satisfies two properties: EK and Separable. An ID-KEM scheme is said to have the *EK* property if the ciphertext $c = \mathsf{Enc_c}(cp, ID; r)$ produced by the key-encapsulation function $\mathsf{Enc_c}$ does not depend on ID. An ID-KEM scheme is said to be *Separable* if the Setup can be separated into two efficient algorithms $\mathsf{Setup_1}$ and $\mathsf{Setup_2}$ such that the following holds. On input security parameter k, $\mathsf{Setup_1}(k)$ returns a string cp_1, and on input cp_1, $\mathsf{Setup_2}(cp_1)$ returns a master key mk and a second string cp_2. The system parameter string is $cp = (cp_1, cp_2)$, and we require that the ciphertext $c = \mathsf{Enc_c}((cp_1, cp_2), ID; r)$ produced by the key-encapsulation function $\mathsf{Enc_c}$ does not depend on cp_2.

Strong ID-One-Wayness. Following the definition in [2], a basic security requirement for ID-KEM schemes is *ID-One-Wayness* (ID-OW). For constructing UDVS schemes, we need a stronger requirement that we call *Strong ID-One-Wayness* (ST-ID-OW). An ID-KEM scheme is said to have the ST-ID-OW property if it is infeasible for an attacker A to win the following two-stage game. In Stage 1, A is given the system pars. cp and outputs a recipient identity ID he wants to be challenged on. In Stage 2, A is given a random KEM challenge ciphertext $c = \mathsf{Enc_c}(cp, ID; r)$ intended for recipient ID but we allow A to adaptively 'change his mind' about the challenge identity; at the end of Stage 2, A outputs an identity ID^* and an estimate $\widehat{K}^*$ for the decryption $K^* = \mathsf{Decrypt}(cp, sk_{ID^*}, c)$ of c under secret-key sk_{ID^*} corresponding to identity ID^*. A is said to win if $\widehat{K}^* = K^*$ (in both stages, A is allowed to query any $ID' \neq ID^*$ to the $\mathsf{Extract}$ oracle). Note that in the weaker ID-OW notion [2] A is not able to change the identity picked at the end of Stage 1.

We remark that the Boneh-Franklin IBE [2] can seen as derived from an underlying separable EK ID-KEM, whereas the Cocks IBE scheme [14] does not seem to give rise to such a KEM.

Constructing a UDVS Scheme from a Separable EK ID-KEM Scheme. We can now describe our general construction of a UDVS scheme from a Separable EK ID-KEM scheme which achieves strong ID-OneWayness.

1. **Com. Par. Generation** GC. Compute $cp_1 = \mathsf{Setup_1}(k)$. The common parameters are cp_1.
2. **Signer Key Generation** GKS. Given common parameters cp_1, compute $(cp_2, mk) = \mathsf{Setup_2}(cp_1)$. The public key is (cp_1, cp_2). The secret key is (cp_1, cp_2, mk).
3. **Verifier Key Generation** GKV. Given common parameters cp_1, let ID_0 and cp_{20} denote any fixed strings. Compute KEM ciphertext $c = \mathsf{Enc_c}((cp_1, cp_{20}), ID_0; r_c)$ for uniformly random $r_c \in S_R$. The public key is c. The secret key is r_c.
4. **Signing** S. Given the signer's secret key (cp_1, cp_2, mk), and message m, compute $sk_m = \mathsf{Extract}((cp_1, cp_2), mk, m)$. The PV signature is sk_m.
5. **Public Verification** V. Given the signer's public key (cp_1, cp_2) and a message/PV-sig. pair (m, sk_m), compute a random KEM ciphertext to identity string m as $\widehat{c} = \mathsf{Enc_c}((cp_1, cp_2), m; \widehat{r})$ for uniformly random $\widehat{r} \in S_R$ with associated encapsulated key $\widehat{K} = \mathsf{Enc_K}((cp_1, cp_2), m; \widehat{r})$. Accept if and only if $\widehat{K}' = \widehat{K}$, where $\widehat{K}' = \mathsf{Decrypt}((cp_1, cp_2), sk_m, \widehat{c})$.
6. **Designation** CDV. Given the signer's public key (cp_1, cp_2), verifier's public key c and a message/PV-sig. pair (m, sk_m), compute $K_{c,m} = \mathsf{Decrypt}((cp_1, cp_2), sk_m, c)$. The DV signature is $K_{c,m}$.
7. **Designated Verification** VDV. Given a signer's public key (cp_1, cp_2), a verifier's secret key r_c, and message/DV-sig. pair $(m, K_{c,m})$, compute $\widehat{K}_{c,m} = \mathsf{Enc_K}((cp_1, cp_2), m; r_c)$ and accept if and only if $\widehat{K}_{c,m} = K_{c,m}$.

The underlying idea behind the construction is a correspondence between the ID-KEM setting and the UDVS setting, where one can make associations between: signer and PKG, messages and identities, DV-sigs. and session keys, designator and decryptor, verifier and encryptor. We point out however the reasons behind the necessity of the special requirements on the ID-KEM scheme: (1) The DV-Consistency of the UDVS scheme translates to the requirement on the ID-KEM scheme that if $c = \mathsf{Enc_c}((cp_1, cp_{20}), ID_0; r_c)$ then $\mathsf{Decrypt}((cp_1, cp_2), sk_{ID}, c) = \mathsf{Enc_K}((cp_1, cp_2), ID; r)$ for *any* ID and the corresponding secret key sk_{ID} to ID. This requirement is satisfied by all Separable EK ID-KEM schemes, but not for general ID-KEM schemes. (2) The ID-KEM separability property is necessary in order that the verifier key-generation algorithm GKV does not need the signer's public key pk_1 — we require a UDVS scheme to allow verifiers to generate keys just once, not once per signer. (3) The ID-KEM needs to have the *strong* ID-OneWayness to ensure the *existential* DV unforgeability for the constructed UDVS scheme.

Constructing an ID-KEM from a UDVS scheme. Interestingly, the above correspondence can also be used in the other direction to construct an ID-KEM scheme (and hence an IBE scheme) from any DV-unforgeable UDVS scheme which is DVSig-Unique and achieves perfect unconditional privacy. The latter properties are needed for the consistency of the ID-KEM construction. The ID-KEM construction is as follows (we let F denote the universal forgery algorithm associated with the UDVS scheme, which exists by Lemma 1).

1. **System Par. Gen.** Setup. Given security parameter k, compute $cp = \mathsf{GC}(k)$ and $(sk_1, pk_1) = \mathsf{GKS}(cp)$. The system parameters are (cp, pk_1). The master key is (cp, sk_1).
2. **Secret-Key Extraction** Extract. Given master key sk_1 and identity ID, compute $\sigma_{ID} = \mathsf{S}(sk_1, ID)$. The identity secret key is σ_{ID}.
3. **KEM Encryption** Encrypt. Given system par. (cp, pk_1), identity ID and random input r, compute $(sk_3, pk_3) = \mathsf{GKV}(cp; r)$ using random input r and DV-sig. forgery $\widehat{\sigma}_{ID,pk_3} = \mathsf{F}(cp, pk_1, sk_3, pk_3, ID)$. The KEM ciphertext is pk_3. The encapsulated key is $\widehat{\sigma}_{ID,pk_3}$.
4. **Decryption** Decrypt. Given system par. (cp, pk_1), secret key σ_{ID} corresponding to identity ID, and KEM ciphertext pk_3, compute DV-sig. $\widehat{\sigma}'_{ID,pk_3} = \mathsf{CDV}(pk_1, pk_3, ID, \sigma_{ID})$. The decrypted encapsulated key is $\widehat{\sigma}'_{ID,pk_3}$.

We summarise our equivalence result in the following statement.

Theorem 3 (Equivalence between subclasses of ID-KEM and UDVS Schemes). *(1) Given any separable and EK ID-KEM scheme* $\mathsf{KEM} = (\mathsf{Setup}_1, \mathsf{Setup}_2, \mathsf{Extract}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *which is consistent and achieves Strong ID-One-Wayness (ST-ID-OW notion), we can construct a UDVS scheme which is consistent and DVSig-Unique and achieves complete perfect unconditional privacy (PR notion) and DV-unforgeability (UF-DV notion).*

(2) Conversely, given any UDVS scheme $\mathsf{DVS} = (\mathsf{GC}, \mathsf{GKS}, \mathsf{GKV}, \mathsf{S}, \mathsf{V}, \mathsf{CDV}, \mathsf{VDV}, \mathsf{P}_{\mathsf{KR}})$ *(where P_{KR} is the direct key-reg. protocol) which is DVSig-Unique, consistent, and achieves complete perfect unconditional privacy (PR notion) and DV-unforgeability (UF-DV notion), we can construct an EK ID-KEM scheme* KEM *which is consistent and achieves Strong ID-One-Wayness (ST-ID-OW notion).*

6 Implementation Aspects and Extensions

6.1 Practical Efficiency of UDVS scheme DVSBM

Currently, the only known way to construct bilinear group-pairs in which BDH is hard is to set G_1 and G_2 to be subgroups of the group of points on certain elliptic curves, as described in [23, 2, 4, 1]. As shown in [1], for such implementations it is possible to evaluate the bilinear map in less than 20ms on a 1GHz P-III processor. Thus we believe that such potential implementations of our scheme are quite practical for many applications of UDVS schemes. Compared to the ring signature in [3], which can also function as a UDVS scheme when restricted to Two-Users as mentioned in Section 1.1, our scheme requires only a single pairing evaluation for designated verification (plus an exponentiation) whereas the scheme in [3] requires three pairing evaluations for this purpose. On the other hand, the scheme in [3] requires only two exponentiations for designation, which may be more efficient than the one pairing evaluation for designation in our scheme.

6.2 Achieving Unforgeability against the KRA

One may require unforgeability of DV-sigs. even against the KRA, which is a stronger than DV-unforgeability notion we defined. To achieve this one can replace the direct key-reg. protocol that we assumed by a zero-knowledge proof of knowledge of the verifier's secret-key. For the scheme DVSBM, the Schnorr proof of knowledge of discrete-logs protocol [29] should suffice for this purpose, although we do not claim a formal proof of security for the resulting scheme.

6.3 Communication-Efficient Selective Disclosure for UDVS Scheme DVSBM

In the application of UDVS schemes to certification systems, Alice's certificate may contain n statements, but Alice may wish to further protect her privacy by disclosing only a *selected subset* of $r < n$ signed statements to Bob. This is easily achieved if Alice obtains a separate signature from the CA for each statement, but requires Alice to send (and designate) r signatures to Bob. Here we observe that for the scheme DVSBM, Alice can reduce the communication cost to only a single DV signature length (and also reduce her computation cost to only one designation and $r-1$ group operations) by using similar techniques as used in [3]. Namely, given the PV signatures $(\sigma_1, \ldots, \sigma_r)$ by a signer with public key $y_1 = g_1^{x_1}$ on messages $(m_1, \ldots, m_r)$, with $\sigma_i = h_i^{x_1}$ and $h_i = H(m_i)$ for $i = 1, ..., r$, a user who wishes to designate a signature on these messages to a verifier with public key $y_3 = g_1^{x_3}$, first multiplies the PV signatures to get $\sigma = \sigma_1 \cdots \sigma_r$, and then designates the product to get $\sigma_{dv} = e(y_3, \sigma)$. The verifier receives $(m_1, \ldots, m_r)$, y_1 and σ_{dv}, computes $\widehat{\sigma}_{dv} = e(y_1^{x_3}, h)$ where $h = h_1 \cdots h_r$ and checks that $\widehat{\sigma}_{dv} = \sigma_{dv}$. The scheme can be proved DV-unforgeable in the 'aggregate signature' sense defined in [4] by reduction from the DV-unforgeability of DVSBM.

7 Conclusions and Future Work

We introduced *Universal Designated-Verifier Signature* (UDVS) schemes to improve the privacy of users in certification systems while maintaining the ease of use of electronic certificates. We defined precise security notions for UDVS schemes, proposed an efficient deterministic UDVS scheme based on bilinear group-pairs, and proved that our scheme achieves our desired security notions (in the random-oracle model), assuming the hardness of the Bilinear Diffie Hellman problem for the underlying group-pair. We also showed a general relationship between UDVS schemes and ID-Based encryption schemes, and discussed extensions to our basic scheme. In [26], we extend this work and show how to construct practical randomized UDVS schemes based on the classical Diffie-Hellman and RSA problems, in the random-oracle model. Threshold versions of UDVS schemes, in which the designator or designated-verifier consist of groups of users, are an interesting topic for future research. Another interesting problem is to construct a practical UDVS scheme which is unforgeable in the *standard* computational model with respect to established cryptographic assumptions.

Acknowledgments. The work of Ron Steinfeld, Huaxiong Wang and Josef Pieprzyk was supported by ARC Discovery Grant DP0345366. Huaxiong Wang's work was also supported in part by ARC Discovery Grant DP0344444.

References

1. P. Barreto, H. Kim, B. Lynn, and M. Scott. Efficient Algorithms for Pairing-based Cryptosystems. In *Crypto 2002*, volume 2442 of *LNCS*, pages 354–368, Berlin, 2002. Springer-Verlag.
2. D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. In *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229, Berlin, 2001. Springer-Verlag.
3. D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. In *Eurocrypt 2003*, LNCS, Berlin, 2003. Springer-Verlag. (To Appear).
4. D. Boneh, B. Lynn, and H. Shacham. Short Signatures from the Weil Pairing. In *Asiacrypt 2001*, volume 2248 of *LNCS*, pages 514–532, Berlin, 2001. Springer-Verlag. See full updated version available at `http://crypto.stanford.edu/ dabo/pubs.html`.
5. S. Brands. A technical overview of digital credentials, 1999. Available at `http://www.xs4all.nl/ brands/`.
6. S. Brands. *Rethinking Public Key Infrastructures and Digital Certificates.* MIT Press, 2000.
7. Gilles Brassard, David Chaum, and Claude Crpeau. Minimum Disclosure Proofs of Knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, October 1988.
8. J. Camenisch and A. Lysyanskaya. An Efficient System for Non-Transferrable Anonymous Credentials with Optional Anonymity Revocation. In *Eurocrypt 2001*, volume 2045 of *LNCS*, pages 93–118, Berlin, 2003. Springer-Verlag.
9. J. Camenisch and M. Michels. Confirmer Signature Schemes Secure against Adaptive Adversaries. In *Eurocrypt 2000*, volume 1807 of *LNCS*, pages 243–258, Berlin, 2000. Springer-Verlag.
10. D. Chaum. Security without ID: Transaction Systems to Make Big Brother Obsolete. *Communications of the ACM*, 28(10):1030–1044, 1985.
11. D. Chaum. Zero-Knowledge Undeniable Signatures. In *Eurocrypt '90*, volume 473 of *LNCS*, pages 458–464, Berlin, 1991. Springer-Verlag.
12. D. Chaum. Designated Confirmer Signatures. In *Eurocrypt '94*, volume 950 of *LNCS*, pages 86–91, Berlin, 1994. Springer-Verlag.
13. D. Chaum and H. van Antwerpen. Undeniable Signatures. In *Crypto '89*, volume 435 of *LNCS*, pages 212–216, Berlin, 1990. Springer-Verlag.
14. C. Cocks. An Identity Based Encryption Scheme Based on Quadratic Residues. In *Cryptography and Coding 2001*, volume 2260 of *LNCS*, pages 360–363, Berlin, 2001. Springer-Verlag.

15. J-S. Coron. On the Exact Security of Full Domain Hash. In *CRYPTO 2000*, volume 1880 of *LNCS*, pages 229–235, Berlin, 2000. Springer-Verlag.
16. A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions of Identification and Signature Problems. In *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194, Berlin, 1987. Springer-Verlag.
17. R. Gennaro and T. Rabin. RSA-Based Undeniable Signatures. *J. of Cryptology*, 13:397–416, 2000.
18. O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(3):690–728, 1991.
19. S. Goldwasser and S. Micali. Probabilistic Encryption. *J. of Computer and System Sciences*, 28(2):270–299, 1984.
20. S. Goldwasser, S. Micali, and R. Rivest. A Digital Signature Scheme Secure against Adaptively Chosen Message Attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.
21. M. Jakobsson, K. Sako, and R. Impagliazzo. Designated Verifier Proofs and Their Applications. In *Eurocrypt '96*, volume 1070 of *LNCS*, pages 143–154, Berlin, 1996. Springer-Verlag.
22. A. Joux. A one round protocol for tripartite Diffie-Hellman. In *Fourth Algorithmic Number Theory Symposium (ANTS IV)*, volume 1838 of *LNCS*, pages 385–394, Berlin, 2000. Springer-Verlag.
23. A. Joux. The Weil and Tate Pairings as Building Blocks for Public Key Cryptosystems. In *Fifth Algorithmic Number Theory Symposium (ANTS V)*, volume 2369 of *LNCS*, pages 20–32, Berlin, 2002. Springer-Verlag.
24. H. Krawcyk and T. Rabin. Chameleon Signatures. In *NDSS 2000*, 2000. Available at `http://www.isoc.org/isoc/conferences/ndss/2000/proceedings/`.
25. M. Michels and M. Stadler. Generic Constructions for Secure and Efficient Confirmer Signature Schemes. In *Eurocrypt '98*, volume 1403 of *LNCS*, pages 406–421, Berlin, 1998. Springer-Verlag.
26. Authors of this paper, 2003. Work in progress.
27. T. Okamoto. Designated Confirmer Signatures and Public-Key Encryption are Equivalent. In *Crypto '94*, volume 839 of *LNCS*, pages 61–74, Berlin, 1994. Springer-Verlag.
28. R. Rivest, A. Shamir, and Y. Tauman. How to Leak a Secret. In *Asiacrypt 2001*, volume 2248 of *LNCS*, pages 552–565, Berlin, 2001. Springer-Verlag.
29. C. P. Schnorr. Efficient Identification and Signatures for Smart Cards. In *CRYPTO'89*, volume 435 of *LNCS*, pages 239–251, Berlin, 1990. Springer-Verlag.
30. V. Shoup. *A Proposal for an ISO Standard for Public Key Encryption (Version 1.1)*, December 2001. ISO/IEC JTC 1/SC 27.
31. R. Steinfeld, L. Bull, and Y. Zheng. Content Extraction Signatures. In *International Conference on Information Security and Cryptology ICISC 2001*, volume 2288 of *LNCS*, pages 285–304, Berlin, 2001. Springer-Verlag.

A Proofs

A.1 Proof of Theorem 2

We first observe that DVSBM is a DVSig-Unique scheme. This follows immediately from the facts that there is only one secret key $x_1 \in \mathbb{Z}_{|G_1|}$ corresponding to each signer public key $y_1 = g_1^{x_1}$ (since g_1 is a generator), and the signing and designation algorithms are both deterministic. Secondly, we observe that there exists an efficient universal DV signature forgery algorithm F, which on input $(cp, y_1, (x_3, y_3), m^*)$, computes the unique DV signature $\sigma_{dv} = \mathsf{CDV}(cp, y_1, y_3, (x_3, y_3), m^*, \mathsf{S}(cp, x_1, m^*))$ with probability 1. Namely, F simply computes $\widehat{\sigma} = e(y_1^{x_3}, h)$ as done by the DV verification algorithm, which is equal to σ_{dv}, by the DV-Consistency property Eq. (2). The algorithm F runs in time $t_F = O(T_g \cdot \log_2 |G_1|) + T_e$. We now construct the forgery strategy $\widehat{\mathsf{A}_1}$ as in the proof of Lemma 1, where $\widehat{\mathsf{A}_1}$ simply runs A_1 and perfectly simulates its designation queries using F and the appropriate verifier secret keys from corresponding KRA queries of A_1. The run-time of $\widehat{\mathsf{A}_1}$ is the run-time t_1 of A_1 plus the time $q_d \cdot O(t_F + q_k)$ to search KRA queries and run F for each designation query of A_1. All other queries of A_1 are simply forwarded by $\widehat{\mathsf{A}_1}$ to its oracles. This completes the proof. □

A.2 Proof of Theorem 3

Proof of (1). We show that the UDVS scheme DVS constructed from the given separable EK ID-KEM scheme KEM as in Section 5 has all the claimed properties.

Consistency: PV Verifiability. For any $(sk_1, pk_1) = \mathsf{GKS}(cp)$, we have that $sk_1 = (cp_1, cp_2, mk)$. So $\sigma_{pv} = \mathsf{S}(sk_1, m) = \mathsf{Extract}((cp_1, cp_2), mk, m)$ is the user secret-key corresponding to user identity m and hence $\widehat{K}' = \mathsf{Decrypt}((cp_1, cp_2), \sigma_{pv}, \mathsf{Enc_c}((cp_1, cp_2), m; \widehat{r}))$ in V is equal to $\widehat{K} = \mathsf{Enc_K}((cp_1, cp_2), m; \widehat{r})$ by consistency of KEM, so V returns Acc.

Consistency: DV Verifiability. From the definition of GKV we have that $pk_3 = \mathsf{Enc_c}((cp_1, cp_{20}), ID_0; sk_3)$, and using the Separable and EK properties of KEM, we also have $pk_3 = \mathsf{Enc_c}((cp_1, cp_2), m; sk_3)$ for any m. So since $\sigma_{pv} = \mathsf{S}(sk_1, m) = \mathsf{Extract}((cp_1, cp_2), mk, m)$ is the user secret-key corresponding to identity m, it follows from the consistency of KEM that $\widehat{\sigma}_{dv} = K_{c,m} = \mathsf{Decrypt}((cp_1, cp_2), \sigma_{pv}, \mathsf{Enc_c}((cp_1, cp_2), m; sk_3))$ is equal to $\widehat{K}_{c,m} = \mathsf{Enc_K}((cp_1, cp_2), m; sk_3)$ so VDV returns Acc.

DVSig-Uniqueness. Given (cp_1, pk_1, pk_3, m), the DV signature $\sigma_{dv} = \mathsf{Decrypt} pk_1, \sigma_{pv}, pk_3$ is uniquely determined by (cp_1, pk_1, pk_3, m) since σ_{pv} is the secret-key corresponding to identity m and all secret-keys corresponding to a given identity must give identical decryptions of any given ciphertext, to satisfy the the consistency of KEM.

DV-Unforgeability. Given any efficient DV-UF attacker A against DVS with resources (t, q_s) and non-negligible success probability $\mathbf{Succ}^{\text{UF-DV}}_{\mathsf{A},\mathsf{DVS}}(k)$, we construct an efficient ST-ID-OW attacker $\widehat{\mathsf{A}}$ against KEM, which works as follows on input (cp_1, cp_2). Let $ID_0 \in S_{ID}$ be any identity string. In Stage 1, $\widehat{\mathsf{A}}$ just outputs ID_0 as the challenge identity. In Stage 2, $\widehat{\mathsf{A}}$ is given the challenge ciphertext $c^* = \mathsf{Enc_c}((cp_1, cp_2), ID_0; r^*)$ for uniformly random $r^* \in S_R$. Then $\widehat{\mathsf{A}}$ sets $pk_3 = c^*$, $pk_1 = (cp_1, cp_2)$ and runs A on input (cp_1, pk_1, pk_3). When A queries a message m_i to its S oracle, $\widehat{\mathsf{A}}$ forwards it to its $\mathsf{Extract}$ oracle and returns the answer to A. Eventually, A outputs a forgery (m^*, σ^*_{dv}), for a new message m^* never queried by A to be signed and hence never queried by $\widehat{\mathsf{A}}$ to $\mathsf{Extract}$. Then $\widehat{\mathsf{A}}$ outputs (m^*, σ^*_{dv}) as its ID/decrypted-key solution pair. Since $\widehat{\mathsf{A}}$ simulated the view of A exactly as in a UF-DV attack, we know that, with probability at least $\mathbf{Succ}^{\text{UF-DV}}_{\mathsf{A},\mathsf{DVS}}(k)$, we have $\sigma^*_{dv} = \mathsf{Enc}_{\mathsf{K}}(cp_1, cp_2, m^*; r^*)$, which by consistency of KEM is equal to $\mathsf{Decrypt}(cp_1, cp_2, sk_{m^*}, \mathsf{Enc}_{\mathsf{c}}(cp_1, cp_2, m^*; r^*))$, which in turn is equal to $\mathsf{Decrypt}(cp_1, cp_2, sk_{m^*}, c^*)$ by the EK property of KEM, which is the desired output for $\widehat{\mathsf{A}}$ (here sk_{m^*} is the secret key corresponding to identity ID). So $\widehat{\mathsf{A}}$ breaks ST-ID-OW with non-negligible probability $\mathbf{Succ}^{\text{UF-DV}}_{\mathsf{A},\mathsf{DVS}}(k)$, with efficient running time t, and q_s extraction queries, contradicting the assumed ST-ID-OW security of KEM.

Complete Unconditional Privacy. We show the existence of an efficient universal forgery algorithm F, which on input $(cp_1, pk_1, (sk_3, pk_3), m^*)$, computes the unique DV signature $\sigma_{dv} = \mathsf{CDV}(pk_1, pk_3, m^*, \mathsf{S}(sk_1, m^*))$ with probability 1. The claimed complete and unconditional privacy then follows by applying Lemma 1. The forger F computes the forgery as in the DV verification algorithm, i.e. $\widehat{\sigma}_{dv} = \mathsf{Enc}_{\mathsf{K}}(pk_1, m^*; sk_3)$. The algorithm is efficient and is correct with probability 1 due to perfect consistency of KEM, as shown in the proof of the DV-Consistency property.

This completes the proof of part (1).

Proof of (2). We show that the UDVS scheme DVS constructed from the given separable EK ID-KEM scheme KEM as in Section 5 has all the claimed properties.

Consistency. By the assumed privacy of DVS we have from Lemma 1 that the encrypted key $K = \mathsf{F}(cp_1, pk_1, sk_3, pk_3, ID)$ is equal to the decrypted key $K' = \mathsf{CDV}(pk_1, pk_3, ID, \mathsf{S}(sk_1, ID))$ with probability 1, so KEM is consistent.

ST-ID-OW Security. Given any efficient ST-ID-OW attacker A against KEM with resources (t, q_E) and non-negligible success probability $\mathbf{Succ}^{\text{ST-ID-OW}}_{\mathsf{A},\mathsf{KEM}}(k)$, we construct an efficient UF-DV attacker $\widehat{\mathsf{A}}$ against DVS, which works as follows on input (cp, pk_1, pk_3). First, $\widehat{\mathsf{A}}$ runs A on input $cp = (cp, pk_1)$. When A queries an identity ID_i to its $\mathsf{Extract}$ oracle, $\widehat{\mathsf{A}}$ forwards it to its S oracle and returns the answer to A. At the end of its Stage 1, A outputs a challenge identity ID, and $\widehat{\mathsf{A}}$ returns the ciphertext pk_3 to A. At the end of Stage 2, A outputs a solution $(\widehat{ID}, K')$, and $\widehat{\mathsf{A}}$ outputs $(\widehat{ID}, K')$ as its message/DV sig. forgery pair. Since $\widehat{\mathsf{A}}$ simulated the view of A exactly as in a ST-ID-OW attack, we know that, with probability at least $\mathbf{Succ}^{\text{ST-ID-OW}}_{\mathsf{A},\mathsf{KEM}}(k)$, $\widehat{\mathsf{A}}$'s output is equal to decrypted key $\mathsf{Decrypt}(cp, sk_{\widehat{ID}}, pk_3)$ for ciphertext pk_3 with respect to identity $\widehat{ID}$, namely the unique DV Sig. $\sigma^*_{dv} = \mathsf{CDV}(pk_1, pk_3, \widehat{ID}, \mathsf{S}(cp, sk_1, \widehat{ID}))$ on message $\widehat{ID}$, which was not queried by A to $\mathsf{Extract}$, and thus not queried by $\widehat{\mathsf{A}}$ to S. So $\widehat{\mathsf{A}}$ breaks UF-DV of DVS with probability $\mathbf{Succ}^{\text{ST-ID-OW}}_{\mathsf{A},\mathsf{KEM}}(k)$, running time t, and q_E signature queries. This completes the proof of part (2). $\square$