Strong Designated Verifier Signature in a Multi-user Setting

M. Choudary Gorantla **Colin Boyd** **Juan Manuel González Nieto**

Information Security Institute, Faculty of IT, Queensland University of Technology
GPO Box 2434, Brisbane, QLD 4001, Australia.
Email: mc.gorantla@isi.qut.edu.au, {c.boyd,j.gonzaleznieto}@qut.edu.au

Abstract

The security of strong designated verifier (SDV) signature schemes has thus far been analyzed only in a two-user setting. We observe that security in a two-user setting does not necessarily imply the same in a multi-user setting for SDV signatures. Moreover, we show that existing security notions do not adequately model the security of SDV signatures even in a two-user setting. We then propose revised notions of security in a multi-user setting and show that no existing scheme satisfies these notions. A new SDV signature scheme is then presented and proven secure under the revised notions in the standard model. For the purpose of constructing the SDV signature scheme, we propose a one-pass key establishment protocol in the standard model, which is of independent interest in itself.

Keywords. Strong designated verifier signatures, multi-user setting, standard model, one-pass key establishment

1 Introduction

An undeniable signature (Chaum & van Antwerpen 1989) allows a signer to have complete control over her signature by forcing the verification to be carried out only with her interactive cooperation. The signer can reject a signature that she never generated and at the same time cannot deny her own signature. However, if a group of verifiers collude, she may not be able to control who can verify the signature (Desmedt & Yung 1991). Later, Jakobsson (1994) shows that undeniable signatures are vulnerable to blackmailing attacks.

As a solution to these problems, Jakobsson, Sako & Impagliazzo (1996) introduce the concept of designated verifier (DV) signature that convinces only a single verifier, designated by the signer, of the validity of the signature. This is achieved by allowing a designated verifier to produce a simulated signature that is indistinguishable from an original DV signature. Note that if both the signer and the designated verifier in a DV signature scheme can produce perfectly indistinguishable signatures, the DV signature scheme cannot offer non-repudiation. Thus, the origin of a DV signature is kept ambiguous between the signer and designated verifier to a third party even with the knowledge of both signer's and verifier's private keys. Although a DV signature does not necessitate an interaction between the signer and verifier, it requires the verifier to have a public-private key pair.

SDV SIGNATURES. Jakobsson et al. (1996) briefly describe a weaker trust model (stronger security notion) called "strong designated verifier". Informally, this notion says that only the designated verifier should be able to distinguish between the transcripts of a strong DV (SDV) signature generated by a legitimate signer and any other transcripts simulated by a third party. For all the other users these transcripts should remain computationally indistinguishable. Hence, only the designated verifier can verify and get convinced of the validity of an original SDV signature.

We use an example from Saeednia, Kremer & Markowitch (2003) to explain a typical scenario where SDV signatures are applicable. Suppose a public institution initiates a call for tenders asking price quotations for a task to be accomplished. The institution requires the participating companies to digitally sign their quotations in order to make sure that they are authentic. At the same time the companies may not want the institution to take advantage by showing their quotations to other companies (e.g. to get lower quotations). We can employ DV signatures in this scenario with the companies acting as signers and the institution as the designated verifier. However, a malicious participant can passively eavesdrop on the institution's incoming communication and easily make out that the captured DV signatures are not simulated by the institution. As the verification process for DV signatures does not require any secret information, it will be convinced that the signature has been generated by a legitimate signer and can make a quotation to its advantage. Hence, we require the signature to be SDV signature.

Jakobsson et al. suggest that the transcripts of a DV signature should be probabilistically encrypted with the designated verifier's public key to arrive at an SDV signature. However, as argued by Raimondo & Gennaro (2005) for the case of deniable authentication, it is not clear why encryption must be used to build authentication protocols. Moreover, the additional layer of encryption on top of a DV signature clearly makes the resulting SDV signature less efficient than the underlying DV signature. Saeednia et al. (2003) address this issue and propose an efficient SDV signature scheme, without employing public key encryption.

Tso, Okamoto & Okamoto (2005) propose that an SDV signature scheme can be obtained using any one-pass key establishment (OPKE)[1] protocol. However, if the OPKE protocol uses signature-based authenticators, any third party can easily distinguish the resulting SDV signature's transcripts from simulated ones. Hence, there cannot be a generic way of ob-

Copyright ©2009, Australian Computer Society, Inc. This paper appeared at the Australasian Information Security Conference (AISC 2009), Wellington, New Zealand, January 2009. Conferences in Research and Practice in Information Technology, Vol. 98. Ljiljana Brankovic and Willy Susilo, Eds. Reproduction for academic, not-for profit purposes permitted provided this text is included.

[1] One-pass key establishment facilitates two parties to establish a shared secret by transmitting a single message. More details in Section 5

taining SDV signatures from OPKE protocols. But, this approach is worth exploring for a concrete OPKE protocol, particularly if the resulting SDV signature offers better security properties and is more efficient than existing SDV signatures.

Security for SDV signatures. Although Jakobsson et al. (1996) and Saeednia et al. (2003) propose ways of constructing SDV signature schemes, they do not provide formal notions of security for SDV signatures. Laguillaumie & Vergnaud (2004) are the first to propose formal notions of security for SDV signatures. All existing SDV schemes (Saeednia et al. 2003, Laguillaumie & Vergnaud 2004, Tso et al. 2005, Huang, Susilo, Mu & Zhang 2006) have been analyzed under these notions. However, the notions of Laguillaumie and Vergnaud are only for a two-user setting. We show that security of an SDV signature in a two-user setting does not necessarily imply the same in a multi-user setting, which is a more realistic setting. We then argue that even if we extend the notions of Laugillaumie and Vergnaud to a multi-user setting, they do not adequately model the security of SDV signatures.

Contributions. We revise existing notions of security for SDV signatures with appropriate justification. We then show that none of the previously published SDV signature schemes is secure under the revised notions. A concrete SDV signature that is secure under the revised notions is then presented. The main tool in constructing the SDV signature scheme is a newly proposed OPKE protocol. This protocol is proven secure in the standard model and is independent interest in itself. Specifically, our contributions are:

- Revised notions of security for SDV signatures
- Attacks on all existing SDV signature schemes
- A new SDV signature scheme in the standard model
- A one-pass key establishment protocol in the standard model

Organization. Section 2 explains why SDV signatures must be analyzed in a multi-user setting and also justifies the appropriateness of security notions we consider. Section 3 presents new notions of security for SDV signatures and we show that existing schemes are insecure under these notions in Section 4. Section 5 presents a security model for OPKE protocols and a new OPKE protocol secure under this model. A concrete SDV signature with proofs of security in the standard model is proposed in Section 6. Appendix A explains preliminary concepts, while Appendices B and C provide proofs of security.

2 On Existing Notions of Security for SDV Signatures

We first informally describe notions of security that may be used to analyze an SDV signature scheme. Out of these notions, we consider unforgeability, invisibility and non-transferability. Their appropriateness for analyzing SDV signature schemes is justified later. The idea behind this section is to explain the reader that existing notions of security for SDV signatures are not adequate. Formal definitions appear in Section 3.

Unforgeability. Being a public key signature, we require an SDV signature to be existentially unforgeable against chosen message attack (UF-CMA). However, the definition of UF-CMA for an SDV signature differs from that for a normal signature as the designated verifier can also simulate the SDV signature. UF-CMA for SDV signature ensures that nobody other than the signer or the designated verifier can produce a valid SDV signature on a random message. This notion may be suitably called outsider unforgeability.

Invisibility. Invisibility against chosen message attack (IV-CMA) ensures that only the designated verifier can distinguish the transcripts of an SDV signature from a uniformly random element of the signature space. This implies that a receiver cannot verify the validity of an SDV signature unless it is designated to him. Saeednia et al. (2003) suggest that even a signer who does not keep record of her own transcripts should not be able to distinguish between real transcripts and random elements of signature space. This additional restriction is taken into account by allowing the adversary to corrupt the signer. It allows us to model "forward invisibility".

Non-transferability. The notion of non-transferability against chosen message attack (NT-CMA) is originally defined by Steinfeld, Bull, Wang & Pieprzyk (2003) for universal DV signatures and is regarded as a desired notion also for DV signatures (Lipmaa, Wang & Bao 2005). NT-CMA makes the origin of a DV signature ambiguous between the signer and the designated verifier and consequently prevents the designated verifier from convincing a third-party about the same. This notion has not been considered for any existing SDV signature scheme.

Privacy of Signer's Identity. Laguillaumie & Vergnaud (2004) define privacy of signer's identity against chosen message attack (PSI-CMA) as a desired notion of security for SDV signatures. This notion ensures that it is computationally infeasible for anybody without the knowledge of the private key of the signer or designated verifier to determine the origin of an SDV signature generated by a legitimate signer.

Non-delegatability. Lipmaa et al. (2005) define non-delegatability against chosen message attack for DV signatures. This notion demands that it should be computationally infeasible for the signer or designated verifier to delegate the signing or verification capabilities respectively, without disclosing the corresponding private key.

We now describe the necessity of considering the security of SDV signatures in a multi-user setting. The justification for preferring IV-CMA to PSI-CMA and considering NT-CMA for analyzing the SDV signature schemes is then explained.

2.1 UF-CMA in a Multi-user Setting

We show that security under UF-CMA for SDV signatures in a two-user setting does not imply the same in a multi-user setting. We use an example construction of Baek, Steinfeld & Zheng (2007), originally given in the context of signcryption, for this purpose. Let Σ be an SDV signature scheme secure under the UF-CMA notion in a two-user setting and let σ be its signing algorithm's output. We construct another SDV signature scheme Σ' such that the output of signing algorithm of Σ' is $\sigma' = \sigma \| b$, where b is a bit from the signer's private key. The position of b is determined by a function of the designated verifier's public key. In the security model for a two-user setting the adversary is allowed to query the signing oracle using only one receiver's public key. Although the adversary can get a single bit of the signer's private key, Σ' remains UF-CMA secure in a two-user setting. However, in a multi-user setting where the adversary should be allowed to query the signing oracle with different designated verifiers' public keys, all the bits of the signer's private key can be easily recovered. Thus, Σ' becomes trivially forgeable in this setting.

2.2 PSI-CMA in a Multi-user Setting

Laguillaumie & Vergnaud (2004) show that encrypting the transcripts of a DV signature using an IND-CCA2 public key encryption scheme results in an SDV signature secure under PSI-CMA. But, using such encryption is computationally expensive and thus the aim has been to achieve PSI-CMA without doing so. Hence, Laguillaumie and Vergnaud also propose a concrete SDV signature scheme without using public key encryption, which is claimed to satisfy the PSI-CMA notion. The authors also give proofs of security for earlier schemes (Saeednia et al. 2003, Steinfeld, Wang & Pieprzyk 2004) under this notion. These analyzes are done only in a two-user setting. However, we explain below that an SDV signature scheme that does not employ public key encryption cannot satisfy PSI-CMA in a multi-user setting.

Recall the example in Section 1, where a public institution initiates call for tenders. As in any other public key signature, the participating companies, acting as signers, have to send their public key certificate along with the SDV signature so that the institution, acting as designated verifier, can verify the signature. However, if the transcripts are not encrypted a passive adversary can easily identify the origin of the signature, trivially exposing the signer's identity.

It may be assumed that the signer's certificate does not accompany the SDV signature and also that the designated verifier has the certificates of all the signers. However, in this case the designated verifier unnecessarily has to do large number[2] of verifications, which could easily make the scheme far less efficient than a scheme that employs public key encryption. Another possible solution is to consider a seemingly weaker notion that guarantees reasonable security and at the same time can be realized by efficient SDV signature schemes. We follow the later approach and formalize the notion of IV-CMA for analyzing SDV signature schemes.

PSI-CMA vs. IV-CMA. As discussed above, an SDV signature scheme that has signer's public key certificate in the transcripts cannot guarantee PSI-CMA. But, such a scheme can satisfy the IV-CMA notion defined in Section 3.2 (For example, our proposed SDV signature scheme guarantees IV-CMA). On the other hand, it seems that PSI-CMA is a stronger notion than IV-CMA (Galbraith & W.Mao 2003). We leave the task of formally establishing relation between PSI-CMA and IV-CMA for future work.

Forward Invisibility. As stated earlier we model forward invisibility by allowing the adversary to corrupt the signer. We explain the necessity of considering forward invisibility using the example in Section 1. In this scenario, assume that the mutually distrusting companies acting as signers collude among themselves by revealing their private keys to each other. The goal of these companies is to maximize the bid as much as possible and make one of the group members win the contract. The others may receive some share for their cooperation. If the SDV signature used is forward invisible, the colluded members cannot verify the other members' quotations even if their private keys are revealed. On the other hand, if the SDV signature used is not forward invisible, the colluded members can make sure that every other colluded member is bidding to a predetermined value assigned to it. We show in Section 4 that none of the existing schemes is forward invisible.

2.3 The necessity of considering NT-CMA

Laguillaumie & Vergnaud (2004) argue that the notion of PSI-CMA implies NT-CMA and the existing SDV signature schemes (Saeednia et al. 2003, Laguillaumie & Vergnaud 2004, Huang et al. 2006) are formally analyzed only under UF-CMA and PSI-CMA. However, we show that PSI-CMA does not necessarily imply NT-CMA by constructing a simple scheme that satisfies UF-CMA and PSI-CMA but does not guarantee NT-CMA.

Assume that an SDV signature is constructed by encrypting a signer's UF-CMA secure normal signature with the public key of the designated verifier. The resulting SDV signature is UF-CMA secure if an IND-CCA2 encryption scheme is used (An, Dodis & Rabin 2002). We have earlier discussed that if the encryption scheme is IND-CCA2 secure then the resulting SDV signature provides PSI-CMA. Note that PSI-CMA assumes that the adversary does not have either the private key of the signer or that of the designated verifier. This construction clearly does not satisfy NT-CMA. The designated verifier can always show to a third party that the decrypted publicly verifiable signature is one from the original signer. Hence, for an SDV signature scheme to be considered secure it must also be analyzed under NT-CMA.

2.4 Notions Omitted

Lipmaa et al. (2005) and Li, Lipmaa & Pei (2005) observe that many DV signature and SDV signature schemes do not have this property. Lipmaa et al. propose a DV signature scheme that is claimed to formally satisfy this notion. However, as observed by (Kudla (2006), p.81) the proof of non-delegatability for this DV signature scheme gives the adversary the same oracle access and seems to have the same objective as in the unforgeability notion. Moreover, it is not clarified what access the adversary has to the functions of the private keys. It is not possible to formally specify these functions as they may vary from scheme to scheme. Hence, we do not consider non-delegatability for analyzing SDV signatures.

3 Security of SDV Signatures in a Multi-user Setting

An SDV signature scheme is specified by four polynomial time algorithms: common-key-gen, user-key-gen, sign and verify.

common-key-gen: is a probabilistic polynomial time (PPT) algorithm that takes the security parameter k as input and outputs the common/public parameters params used in the scheme. These parameters include description of the underlying groups, hash functions and signature space $\mathcal{S}$ etc.

user-key-gen: is a PPT algorithm that takes params as input and outputs the user's public-private key pair (pk, sk). Note that the same key pair can be used in the sign or verify algorithms depending on the role the user plays in the scheme.

sign: is a PPT algorithm that takes params, signer's private key sk_s, verifier's public key pk_v and a message m as input. An SDV signature $\sigma \in \mathcal{S}$ created for a designated verifier with public key pk_v is returned as output.

[2] In the above example the institution has to do $O(n^2)$ verifications, instead of the much desired n verifications, where n is the number of participants.

verify: is a deterministic polynomial time algorithm that takes params, signer's public key pk_s, verifier's private key sk_v, a message m and an SDV signature σ. It outputs a boolean value *true* if σ is valid SDV signature created on m under sk_s and pk_v. Otherwise *false* is returned.

We assume that all the designated verifiers' public keys are registered with a key registration authority (KRA) in a direct key registration protocol i.e. the KRA checks that the users know their private keys. We now present the desired notions of security for an SDV signature scheme namely, UF-CMA, IV-CMA and NT-CMA in a multi-user setting. Depending on the notion of security, an adversary against an SDV signature scheme is allowed ask the below queries:

The challenger runs the user-key-gen algorithm for n users and generates public-private key pairs for them. Without loss of generality, let (pk_s, sk_s) and (pk_v, sk_v) be the public-private key pairs of a signer and a designated verifier respectively.

sign queries: On a sign query with the input (m, pk_s, pk_v), the adversary $\mathcal{A}$ is given the SDV signature σ computed on the message m using the keys sk_s and pk_v.

verify queries: On a verify(m, σ, pk_s, pk_v), $\mathcal{A}$ is returned an output *true* if σ is a valid SDV signature on m under the keys sk_s and pk_v. Otherwise, *false* is returned.

simulate queries: On a Simulate query with input (m, pk_s, pk_v), a simulated signature σ' computed on the message m using the keys pk_s and sk_v is returned to $\mathcal{A}$.

corrupt queries: On a corrupt query issued on a user U with public-private key pair (pk, sk), the private key sk is returned to $\mathcal{A}$.

3.1 Unforgeability

An adversary $\mathcal{A}^u$ against the UF-CMA notion of an SDV signature is allowed to issue sign, verify and corrupt queries. Finally, it outputs a forgery $(m^*, \sigma^*, pk_s^*, pk_v^*)$. For $\mathcal{A}^u$ to win the game, the following conditions must hold:

1. σ^* is a valid SDV signature i.e. verify$(m^*, \sigma^*, pk_s^*, sk_v^*) =$ *true*
2. σ^* has not been an output of an earlier sign query (sign queries with input (m^*, pk_s, pk_v) for $pk_s = pk_s^*$ and $pk_v = pk_v^*$ are also allowed)
3. there has been no corrupt query issued to the user U_s^* or U_v^* with the public keys pk_s^* or pk_v^* respectively.

The advantage of $\mathcal{A}^u$, $Adv_{\mathcal{A}^u}(k)$, against the UF-CMA notion of SDV signature is given as its probability of winning the above game. We say that an SDV signature scheme is secure under the UF-CMA notion if $Adv_{\mathcal{A}^u}(k)$ is negligible. Note that our notion of UF-CMA for SDV signatures models *strong unforgeability.*

3.2 Invisibility

An adversary $\mathcal{A}^i$ against the IV-CMA notion of an SDV signature is allowed to issue sign, verify and corrupt queries. At the end of stage 1, $\mathcal{A}^i$ outputs a message m^*, a signer's public key pk_s^* and a verifier's public key pk_v^*.

The challenger randomly chooses $b \xleftarrow{R} \{0, 1\}$. If $b = 0$, $\mathcal{A}^i$ is given a valid SDV signature σ^* computed on the message m^* using sk_s^* and pk_v^*. Otherwise, σ^* is chosen uniformly at random from the signature space $\mathcal{S}$ and returned to $\mathcal{A}^i$. $\mathcal{A}^i$ can continue asking the queries in stage 2. Finally, it outputs a bit b'. $\mathcal{A}^i$ wins the IV-CMA game if the following conditions hold:

1. $b' = b$
2. there has been no verify query with the input $(m^*, \sigma^*, pk_s^*, pk_v^*)$
3. the user with the public key pk_v^* remains uncorrupted (the user with the public key pk_s^* may be corrupted at any time)

The advantage of $\mathcal{A}^i$ in winning the above game is:

$$Adv_{\mathcal{A}^i}(k) = |2 \cdot \Pr[b' = b] - 1|$$

We say that an SDV signature scheme is secure under the IV-CMA notion if $Adv_{\mathcal{A}^i}(k)$ is negligible.

3.3 Non-transferability

We say that an SDV signature is NT-CMA secure if there exists a polynomial time algorithm simulate that on input (pk_s, pk_v, sk_v, m) produces a valid SDV signature σ' such that the outputs of the sign and simulate algorithms are indistinguishable. An adversary $\mathcal{A}^n$ against the NT-CMA notion of an SDV signature is allowed to issue simulate queries in addition to sign, verify and corrupt queries. At the end of stage 1, $\mathcal{A}^n$ outputs a message m^* and a signer's public key pk_s^* and a designated verifier's public key pk_v^*. These two users might already have been corrupted. The challenger chooses a bit b. If $b = 0$ it runs the sign algorithm, otherwise runs the simulate algorithm with (m^*, pk_s^*, pk_v^*) as input and produces a signature σ^* and returns it to $\mathcal{A}^n$. $\mathcal{A}^n$ can continue asking queries to the oracles and there is no restriction on the type of query. Finally, $\mathcal{A}^n$ outputs a bit b' and wins the game if $b' = b$.

The advantage of $\mathcal{A}^n$ in winning the above game is:

$$Adv_{\mathcal{A}^n}(k) = |2 \cdot \Pr[b' = b] - 1|$$

We say that an SDV signature scheme is secure under the NT-CMA notion if $Adv_{\mathcal{A}^n}(k)$ is negligible.

Note that this notion allows the adversary to corrupt any user including the users with key pairs (sk_s^*, pk_s^*) and (sk_v^*, pk_v^*). Hence, a scheme secure under the NT-CMA notion guarantees perfect non-transferability.

4 The Security of Existing Schemes

Before presenting our generic construction and its instantiation, we first show that the previously published SDV signature schemes do not satisfy the IV-CMA notion defined in Section 3.2. Particularly, we show that these schemes do not have forward invisibility.

4.1 Deterministic SDV Signature Schemes

The SDV signature schemes of Tso et al. (2005), Huang et al. (2006) and Bhaskar, Herranz & Laguillaumie (2006) depend on the static Diffie-Hellman key between the signer and designated verifier. In these schemes, if the private key of the signer is revealed to $\mathcal{A}^i$, it can compute the static shared key. With this key $\mathcal{A}^i$ can easily win the IV-CMA game by checking whether the challenge SDV signature is real or random. Similarly, the ID-based short SDV signature of Huang et al. (2006), which depends on the ID-based non-interactive key sharing (Sakai, Ohgishi & Kasahara 2000), can be shown insecure under the IV-CMA.

It should also be noted that these schemes (Tso et al. 2005, Huang et al. 2006, Bhaskar et al. 2006) have deterministic signing algorithms and thus produce the same SDV signature between two users on a given message. However, the existing notion PSI-CMA for two-user setting defined by Laguillaumie & Vergnaud (2004) requires the signing algorithm to be a randomized one. Hence, although these schemes are claimed to be secure under PSI-CMA, they are actually not. As PSI-CMA and IV-CMA (without forward invisibility) are equivalent in the two-user setting, they can't be secure under the IV-CMA notion even in the two-user setting. These schemes can nevertheless be proven secure in a weaker notion. However, it implies that once the adversary knows that a given SDV signature is generated by a signer, the next signature by the signer on the same message cannot be secure under PSI-CMA.

4.2 Saeednia et al. 2003

The public parameters in this scheme are (p, q, g, h), where p is a large prime, q is a prime factor of $p-1$, $g \in Z_p^*$ is a generator of order q and h is a one-way function that outputs values in Z_q. The public-private key pairs of users $\hat{A}$ and $\hat{B}$ are $(A = g^a, a)$ and $(B = g^b, b)$ respectively.

The SDV signature generated by $\hat{A}$ for $\hat{B}$ on a message m is (r, s, t). The signer computes this signature by selecting two random values $k \in Z_q$, $t \in Z_q^*$ and calculating $c = B^k \mod p$, $r = h(m, c)$ and $s = kt^{-1} - ra \mod q$. If the signer's private key a is revealed, $\mathcal{A}^i$ can decide whether a given challenge SDV signature is real or random as follows. Let (r^*, s^*, t^*) be the challenge SDV signature, $\mathcal{A}^i$ can compute k^* as $k^* = (s^* + r^*a)t^* \mod q$ and then $c^* = B^{k^*} \mod p$. If $r^* = h(m, c^*)$, $\mathcal{A}^i$ correctly guesses the challenge SDV signature as real, otherwise as random. Note that $\mathcal{A}^i$ wins this game with probability 1 by reconstructing the SDV signature.

4.3 Laguillaumie and Vergnaud 2004

The public parameters are $(q, \mathbb{G}_0, P_0, \mathbb{G}_1, P_1, \mathbb{G}_T, e, h)$, where q is a prime number, $\mathbb{G}_0$, $\mathbb{G}_1$ and $\mathbb{G}_T$ are groups of order q, P_0 and P_1 are generators of $\mathbb{G}_0$ and $\mathbb{G}_1$ respectively, $e : \mathbb{G}_0 \times \mathbb{G}_1 \to \mathbb{G}_T$ is an admissible bilinear map (Boneh & Franklin 2001) and h is a hash function that maps to $\mathbb{G}_1$. Let $(P_A = aP_0, a)$ and $(P_B = bP_0, b)$ be the public-private key pairs of $\hat{A}$ and $\hat{B}$ respectively.

The SDV signature of $\hat{A}$ on the message m for the user $\hat{B}$ is (r, s), where r is a randomly selected string and s is computed as $s = e(P_B, aH)$ for $H = h(m, r)$. Note that in this scheme the random seed is also part of the SDV signature that is sent over a public channel. Hence, in this scheme if $\mathcal{A}^i$ corrupts the signer and obtains the private key a, it can win the IV-CMA game by reconstructing the SDV signature. Given a challenge SDV signature (r^*, s^*), $\mathcal{A}^i$ computes $H^* = h(m, r^*)$ and checks if $s^* = e(P_B, aH^*)$. If so it correctly guesses the SDV signature as real, otherwise as random.

4.4 Susilo et al. 2004

This is an ID-based scheme that uses symmetric bilinear pairings. The system parameters are $(q, \mathbb{G}_1, \mathbb{G}_T, P, e, H_0, H_1, P_{pub})$, where q is a prime number, $\mathbb{G}_1$ and $\mathbb{G}_T$ are groups of order q, P is a generator of $\mathbb{G}_1$, $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$ is an admissible bilinear map and H_0 ,H_1 are two hash functions that map strings of arbitrary length to $\mathbb{G}_1$ and $\mathbb{Z}_q$ respectively. P_{pub} is the public key of the key generation center set as $P_{pub} = sP$, where s is its master secret. Let $(Q_A = H_0(\hat{A}), S_A = sQ_A)$ and $(Q_B = H_0(\hat{B}), S_B = sQ_B)$ be the public-private key pairs of $\hat{A}$ and $\hat{B}$ respectively.

To sign a message m for $\hat{B}$, $\hat{A}$ selects two random values $k \in \mathbb{Z}_q$ and $t \in \mathbb{Z}_q^*$ and computes $c = e(Q_B, P)^k$, $r = H_1(m, c)$ and $T = t^{-1}kP - rS_A$. The SDV signature is (T, r, t). Let the challenge SDV signature in the IV-CMA game be (T^*, r^*, t^*). $\mathcal{A}^i$ obtains S_A by corrupting $\hat{A}$. This scheme can be seen as an extension of Saeednia et al. (2003) scheme to the ID-based setting. However, the attack presented on Saeednia et al. does not directly apply to this scheme as the random seed k^* cannot be extracted by $\mathcal{A}^i$ from the challenge SDV signature. Nevertheless, we show that $\mathcal{A}^i$ can still win the IV-CMA game. $\mathcal{A}^i$ sets $K^* = t^*(T^* + r^*S_A)$ and then $c^* = e(Q_B, K^*)$. It guesses the SDV signature as real if $r^* = H_1(m^*, c^*)$, otherwise as random.

5 One-pass Key Establishment

One-pass (authenticated) key establishment (OPKE) facilitates two parties to establish a shared secret by transmitting a single message. OPKE protocols are useful in non-interactive applications like E-mail systems, which do not demand the recipient to be always online. Although OPKE protocols cannot offer the same level of security as two-pass/multi-pass protocols, they provide the right trade-off between security and efficiency. We present a OPKE protocol by a simple modification to the one-round key establishment protocol of Jeong, Katz & Lee (2004). The protocol is proven secure under the popular Bellare-Rogaway models (Bellare & Rogaway 1994, Bellare, Pointcheval & Rogaway 2000) after suitably adapting them to the one-pass case. This protocol is the first OPKE protocol in the standard model with a proof of security. It also turns out to be more efficient than existing OPKE protocols in the random oracle model (Krawczyk 2005, Ustaoglu 2008), when the static Diffie-Hellman key between a pair of users is precomputed.

5.1 Security Model for One-pass key establishment

To analyze the security of OPKE protocols, we use the standard models of security for key establishment (Bellare & Rogaway 1994, Bellare et al. 2000) with a few modifications to suit OPKE protocols. As in the earlier models a protocol π is modeled as a collection of n programs running at different parties, $P_1, \ldots, P_n$. Each invocation of π within a party is defined as a *session*, and each party may have multiple sessions running concurrently. The communications network is controlled by an adversary $\mathcal{A}^\pi$, which schedules and mediates all sessions between the parties. $\mathcal{A}^\pi$ is responsible for transmitting messages between parties, and may fabricate or modify messages when desired. $\mathcal{A}^\pi$ activates a party by sending appropriate protocol messages depending on the role $\mathcal{A}^\pi$

wants the party to play. In a OPKE protocol a party can be either an initiator or a responder. Upon activation, the parties perform some computations and update their internal state as per the protocol. The session ID sid is formed by concatenating all protocol messages with the identities of the participating parties. sid is assumed to be unique amongst all sessions between P_i and P_j. Two sessions are said to be *matching sessions* if their session IDs are identical. $\mathcal{A}^\pi$ can perform the following queries.

- $\mathsf{send}(P_i, P_j, \mathsf{sid}, \mathit{role})$. A send query of the form $\mathsf{send}(P_i, P_j, \mathsf{sid}, I)$ activates a party P_i, where sid is the session ID and I is the role *initiator*. In response to this request P_i outputs a message m intended for party P_j. A send query of the form $\mathsf{send}(P_j, P_i, \mathsf{sid}, m, R)$ activates a party P_j with an incoming message m, where R is the role *responder.*
- $\mathsf{corrupt}(P_i)$. With this query $\mathcal{A}^\pi$ learns the static private key of the party P_i. From this point on, $\mathcal{A}^\pi$ may issue any message in which P_i is specified as the sender and play the role of P_i.
- $\mathsf{reveal}(P_i, \mathsf{sid})$. This query returns the unexpired session key (if any) accepted by P_i during a given session sid.
- $\mathsf{test}(P_i, \mathsf{sid})$. To respond to this query, a random bit b is selected. If $b = 0$ then the session key is output. Otherwise, a random key chosen from the probability distribution of session keys is returned.

$\mathcal{A}^\pi$ is allowed to continue with its execution by issuing the above queries even after the test query. Finally, it terminates by outputting its guess b' on distinguishing the session key from a random string.

We first define the notion of *fresh session* for OPKE protocols that is central to the security definition.

Definition 1 (Fresh Session)**.** *Let* sid *be a session completed at a party* P_i *and let* sid^* *be the matching session at* P_j *(there may be no such* sid^**). A session* sid *is said to be fresh if none of the following conditions holds:*

- *If* $\mathcal{A}^\pi$ *makes a* $\mathsf{reveal}(P_i, \mathsf{sid})$ *or a* $\mathsf{reveal}(P_j, \mathsf{sid}^*)$ *(if* sid^* *exists)*
- *If* P_i *is the initiator then* $\mathcal{A}^\pi$ *makes either* $\mathsf{corrupt}(P_i)$ *query before a* send *query containing* sid *or* $\mathsf{corrupt}(P_j)$
- *If* P_i *is the responder then* $\mathcal{A}^\pi$ *makes either* $\mathsf{corrupt}(P_i)$ *or* $\mathsf{corrupt}(P_j)$

$\mathcal{A}^\pi$ wins the above game if it keeps the test session fresh till the end of its execution and if $b' = b$. The advantage of $\mathcal{A}^\pi$ is defined to be $Adv_{\mathcal{A}^\pi} = |\Pr[b' = b] - \frac{1}{2}|$

Definition 2. *A one-pass protocol* π *is secure if the following conditions hold:*

- *When two honest parties complete protocol execution, they both should compute the same session key with all but negligible probability.*
- *If* $Adv_{\mathcal{A}^\pi}$ *is negligible in the security parameter.*

Note that the definition of security takes care of forward secrecy for the initiator i.e. the disclosure of the static private key of an initiator does not compromise the session keys established using that private key in the previous sessions. However, to guarantee initiator's forward secrecy, the adversary needs to remain passive during the test session (Krawczyk 2005).

Recently, Ustaoglu (2008) propose a notion of fresh session for OPKE protocols in the eCK model (LaMacchia, Lauter & Mityagin 2007). The difference between our notion and that of Ustaoglu is that we do not consider session state reveal queries. It is important to consider session state reveal queries in a multi pass protocol, where a party has to keep the session state until it receives appropriate protocol messages from other participant and then to compute the session key. On the other hand, the session state in a OPKE protocol can be securely erased after the sender computes the session key, which can be done without waiting for any protocol messages. Hence, session state reveal queries need not be considered for OPKE protocols.

5.2 A One-pass Key Establishment Protocol

Jeong et al. (2004) propose a one-round two-party protocol called $\mathcal{TS}3$ and prove its security in the standard model. We present a one-pass version of $\mathcal{TS}3$ and prove it secure in the standard model. Note that one can also obtain one pass versions from the protocols of Okamoto (2007) and Boyd, Cliff, Nieto & Paterson (2008) and use them to construct SDV signatures. But, we consider Jeong, Katz and Lee's protocol for efficiency reasons.

Let k be the security parameter, p be large prime number and q be a prime factor of $p-1$. Let $g \in \mathbb{Z}_p^*$ be a generator of order q and let $\mathbb{G}$ be the group generated by g. Let $(x_i, Y_i = g^{x_i})$ be the private-public key pair of a party P_i where $x_i \in \mathbb{Z}_q$. Similarly the private-public key pair of P_j is $(x_j, Y_j = g^{x_j})$. Let $\mathcal{M}$ be a MAC scheme. The below two phases described the OPKE protocol executed between two users P_i and P_j acting as initiator and responder respectively.

Key Establishment P_i computes the static Diffie-Hellman key between P_i and P_j as $K_{i,j} = Y_j^{x_i}$, which will be used for keying the MAC. It chooses $u \overset{R}{\leftarrow} \mathbb{Z}_q$, computes $U = g^u$ and $\tau = \mathcal{M}_{K_{i,j}}(i\|j\|U)$ and sends (U, τ) to P_j.

Key Computation P_i computes the session key as $\kappa_i = Y_j^u$. The party P_j first checks that the received tag is valid using the static Diffie-Hellman key computed as $K_{j,i} = Y_i^{x_j}$. It then computes the session key $\kappa_j = U^{x_j}$.

Note that we have used the group elements as the key for the MAC and also as the session key instead of a random string. A simple way deriving key when $\mathbb{G} \subset \mathbb{Z}_p^*$ is just by selecting the k most significant bits or the k least significant bits of the elements of $\mathbb{G}$ (Fouque, Pointcheval, Stern & Zimmer 2006). Generic methods for implementing a suitable key derivation function are given by Dodis, Gennaro, Håstad, Krawczyk & Rabin (2004) and Chevassut, Fouque, Gaudry & Pointcheval (2005).

Theorem 1. *Let* $\mathcal{A}^\pi$ *be an adversary against the proposed OPKE protocol. Then the advantage of* $\mathcal{A}^\pi$ *against the security of the protocol (with initiator's forward secrecy as defined in Section 5.1) is* $\mathsf{Adv}_{\mathcal{A}^\pi}(k) \leq 2n_s(2 \cdot \mathsf{AdvDDH}_{\mathcal{A}^{ddh}}(k) + \mathsf{AdvMac}_{\mathcal{F}}(k)) + \frac{2n_s}{q}$.

where n_s *is upper bound on the number of sessions* $\mathcal{A}^\pi$ *may activate,* $\mathsf{AdvDDH}_{\mathcal{A}^{ddh}}(k)$ *is the advantage a polynomial time algorithm* $\mathcal{A}^{ddh}$ *in solving the DDH problem in* $\mathbb{G}$ *and* $\mathsf{AdvMac}_{\mathcal{F}}(k)$ *is the success probability of a forger* $\mathcal{F}$ *against the SUF-CMA notion of the MAC.*

The proof of Theorem 1 is in Appendix B.

6 A Concrete SDV Signature Scheme

We now present an SDV signature scheme based on the OPKE protocol proposed in Section 5. We obtain the new SDV signature by using a message authentication code along with the OPKE protocol. The session key obtained in the protocol is used to compute a tag on the given message. The protocol transcripts together with the computed tag form the an SDV signature. The verification can be done by first computing the session key and then using it to verify the tag on the input message. Below we present the description of the scheme executed between a signer P_i and a designated verifier P_j.

common-key-gen: This algorithm takes the security parameter k as input and generates the system parameters $\mathsf{params} = (p, q, \mathbb{G}, g, \mathcal{M}, \mathcal{S})$, where p is a large prime number, q be a prime factor of $p-1$, $g \in \mathbb{Z}_p^*$ is a generator of order q generating the group $\mathbb{G}$ $\mathcal{M}$ is a MAC and $\mathcal{S}$ is the signature space.

user-key-gen: This algorithm takes the security parameter k and params as input. The public-private key pair of a user is generated in the same way as in the the OPKE protocol. Let $(y_i = g^{x_i}, x_i)$ and $(y_j = g^{x_j}, x_j)$ be the the public-private key pairs of P_i and P_j respectively, where $x_i, x_j \in \mathbb{Z}_q$.

sign: Let $m \in \{0,1\}^*$ be the input message. The signer P_i executes the OPKE protocol and generates the SDV signature $\sigma = (U, \tau_1, \tau_2)$ as follows: Let (U, τ_1) be the outgoing message of the protocol and let κ be the session key established. P_i computes the tag $\tau_2 = \mathcal{M}_\kappa(m)$.

verify: On receiving a purported SDV signature (U, τ_1, τ_2) on a message m, a designated verifier P_j does the following: P_j first executed the OPKE protocol on the incoming message (U, τ_1) and computed the session key κ. It now accepts the SDV signature only if τ_2 is a valid tag on the message m under the session key κ.

Security of the new SDV signature scheme. We show that the proposed SDV signature signature scheme is secure under the UF-CMA, IV-CMA and NT-CMA notions defined in Section 3. Specifically we prove the following theorems in Appendix C.

Theorem 2. *Let $\mathcal{A}^u$ be an adversary against the UF-CMA notion of the SDV signature. Then the advantage of $\mathcal{A}^u$ is* $\mathsf{Adv}_{\mathcal{A}^u}(k) \leq n(n-1)\left(\mathsf{AdvMac}_{\mathcal{F}}(k) + \mathsf{Adv}_{\mathcal{A}^\pi}(k)\right)$.

Theorem 3. *Let $\mathcal{A}^i$ be an adversary against the IV-CMA notion of the SDV signature. Then the advantage of $\mathcal{A}^i$ is* $\mathsf{Adv}_{\mathcal{A}^i}(k) \leq 2(\mathsf{AdvMac}_{\mathsf{prf}}(k) + \mathsf{Adv}_{\mathcal{A}^\pi}(k) - \frac{1}{2^s})$.

Theorem 4. *Let $\mathcal{A}^n$ be an adversary against the NT-CMA notion of the SDV signature. Then the advantage of $\mathcal{A}^n$ is* $\mathsf{Adv}_{\mathcal{A}^n}(k) = 0$.

7 Conclusion

We have shown that the existing security notions for SDV signature are not adequate. We have then proposed new notions of security in the multi-user setting and all existing SDV signature schemes turned out to be insecure under these revised notions. We have then presented a concrete SDV signature scheme and proven its security in the standard model. A new one-pass key establishment protocol used to construct the SDV signature scheme is of independent interest in itself.

Acknowledgements

This work was supported by the Australian Research Council under grant DP0773348.

References

An, J., Dodis, Y. & Rabin, T. (2002), On the Security of Joint Signature and Encryption, *in* 'Advances in Cryptology–EUROCRYPT'02', Vol. 2332 of *LNCS*, Springer, pp. 83–107.

Baek, J., Steinfeld, R. & Zheng, Y. (2007), 'Formal Proofs for the Security of Signcryption.', *Journal of Cryptology* **20**(2), 203–235.

Bellare, M., Kilian, J. & Rogaway, P. (1994), The security of the cipher block chaining message authentication code, *in* 'Advances in Cryptology–CRYPTO '94', Vol. 839 of *LNCS*, Springer, pp. 341–358.

Bellare, M., Pointcheval, D. & Rogaway, P. (2000), Authenticated Key Exchange Secure against Dictionary Attacks, *in* 'Advances in Cryptology–EUROCRYPT'00', Vol. 1807 of *LNCS*, Springer, pp. 139–155.

Bellare, M. & Rogaway, P. (1994), Entity Authentication and Key Distribution, *in* 'Advances in Cryptology–CRYPTO'93', Vol. 773 of *LNCS*, Springer, pp. 232–249.

Bhaskar, R., Herranz, J. & Laguillaumie, F. (2006), Efficient Authentication for Reactive Routing Protocols, *in* '20th International Conference on Advanced Information Networking and Applications–AINA'06', IEEE Computer Society, pp. 57–61.

Boneh, D. & Franklin, M. (2001), Identity-Based Encryption from the Weil Pairing, *in* 'Advances in Cryptology–CRYPTO'01', Vol. 2139 of *LNCS*, Springer, pp. 213–229.

Boyd, C., Cliff, Y., Nieto, J. G. & Paterson, K. G. (2008), Efficient One-Round Key Exchange in the Standard Model, *in* Y. Mu, W. Susilo & J. Seberry, eds, 'Information Security and Privacy, 13th Australasian Conference, ACISP 2008', Vol. 5107 of *LNCS*, Springer, pp. 69–83.

Chaum, D. & van Antwerpen, H. (1989), Undeniable Signatures, *in* 'Advances in Cryptology–CRYPTO'89', Vol. 435 of *LNCS*, Springer, pp. 212–216.

Chevassut, O., Fouque, P.-A., Gaudry, P. & Pointcheval, D. (2005), 'Key Derivation and Randomness Extraction', Cryptology ePrint Archive, Report 2005/061. `http://eprint.iacr.org/`.

Desmedt, Y. & Yung, M. (1991), Weakness of Undeniable Signature Schemes (Extended Abstract), *in* 'Advances in Cryptology–EUROCRYPT'91', Springer, pp. 205–220.

Dodis, Y., Gennaro, R., Håstad, J., Krawczyk, H. & Rabin, T. (2004), Randomness Extraction and Key Derivation Using the CBC, Cascade and HMAC Modes, *in* 'Advances in Cryptology–CRYPTO'04', Vol. 3152 of *LNCS*, Springer.

Fouque, P.-A., Pointcheval, D., Stern, J. & Zimmer, S. (2006), Hardness of Distinguishing the MSB or LSB of Secret Keys in Diffie-Hellman Schemes, *in* M. Bugliesi, B. Preneel, V. Sassone & I. Wegener, eds, 'Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006', Vol. 4052 of *Lecture Notes in Computer Science*, Springer, pp. 240–251.

Galbraith, S. & W.Mao (2003), Invisibility and anonymity of undeniable and confirmer signatures, *in* 'Topics in Cryptology–CT-RSA'03', Vol. 2612 of *LNCS*, Springer-Verlag, pp. 80–97.

Goldwasser, S. & Bellare, M. (1996-2008), 'Lecture Notes on Cryptography'. `http://www.cs.ucsd.edu/~mihir/papers/gb.html`.

Huang, X., Susilo, W., Mu, Y. & Zhang, F. (2006), Short (Identity-Based) Strong Designated Verifier Signature Schemes, *in* 'Information Security Practice and Experience–ISPEC'06', Vol. 3903 of *LNCS*, Springer, pp. 214–225.

Jakobsson, M. (1994), Blackmailing using undeniable signatures, *in* 'Advances in Cryptology–EUROCRYPT'94', Vol. 950 of *LNCS*, Springer, pp. 425–427.

Jakobsson, M., Sako, K. & Impagliazzo, R. (1996), Designated Verifier Proofs and Their Applications, *in* 'Advances in Cryptology–EUROCRYPT'96', Vol. 1070 of *LNCS*, Springer, pp. 143–154.

Jeong, I. R., Katz, J. & Lee, D. H. (2004), One-Round Protocols for Two-Party Authenticated Key Exchange, *in* M. Jakobsson, M. Yung & J. Zhou, eds, 'Applied Cryptography and Network Security, Second International Conference, ACNS 2004', Vol. 3089 of *LNCS*, Springer, pp. 220–232.

Krawczyk, H. (2005), HMQV: A High-Performance Secure Diffie-Hellman Protocol, *in* 'Advances in Cryptology–CRYPTO'05', Vol. 3621 of *LNCS*, Springer, pp. 546–566.

Kudla, C. (2006), Special Signature Schemes and Key Agreement Protocols, PhD thesis, Royal Holloway, University of London.

Laguillaumie, F. & Vergnaud, D. (2004), Designated Verifier Signatures: Anonymity and Efficient Construction from Any Bilinear Map, *in* 'Security in Communication Networks–SCN'04', Vol. 3352 of *LNCS*, Springer, pp. 105–119.

LaMacchia, B., Lauter, K. & Mityagin, A. (2007), Stronger Security of Authenticated Key Exchange, *in* 'Provable Security–ProvSec'07', Vol. 4784 of *LNCS*, Springer, pp. 1–16.

Li, Y., Lipmaa, H. & Pei, D. (2005), On Delegatability of Four Designated Verifier Signatures, *in* 'Information and Communications Security–ICICS'05', Vol. 3783 of *LNCS*, Springer, pp. 61–71.

Lipmaa, H., Wang, G. & Bao, F. (2005), Designated Verifier Signature Schemes: Attacks, New Security Notions and a New Construction, *in* 'Automata, Languages and Programming, 32nd International Colloquium–ICALP'05', Vol. 3580 of *LNCS*, Springer, pp. 459–471.

Okamoto, T. (2007), Authenticated Key Exchange and Key Encapsulation in the Standard Model, *in* K. Kurosawa, ed., 'Advances in Cryptology – ASIACRYPT 2007', Vol. 4833 of *LNCS*, Springer, pp. 474–484.

Raimondo, M. D. & Gennaro, R. (2005), New approaches for deniable authentication, *in* V. Atluri, C. Meadows & A. Juels, eds, 'Proceedings of the 12th ACM Conference on Computer and Communications Security, CCS 2005, Alexandria, VA, USA, November 7-11, 2005', ACM, pp. 112–121.

Saeednia, S., Kremer, S. & Markowitch, O. (2003), An Efficient Strong Designated Verifier Signature Scheme, *in* 'Information Security and Cryptology–ICISC'03', Vol. 2971 of *LNCS*, Springer, pp. 40–54.

Sakai, R., Ohgishi, K. & Kasahara, M. (2000), Cryptosystems based on Pairing, *in* 'SCIS, C20'.

Steinfeld, R., Bull, L., Wang, H. & Pieprzyk, J. (2003), Universal Designated-Verifier Signatures, *in* 'Advances in Cryptology–ASIACRYPT'03', Vol. 2894 of *LNCS*, Springer, pp. 523–542.

Steinfeld, R., Wang, H. & Pieprzyk, J. (2004), Efficient Extension of Standard Schnorr/RSA Signatures into Universal Designated-Verifier Signatures, *in* 'Public Key Cryptography–PKC'04', Vol. 2947 of *LNCS*, Springer, pp. 86–100.

Tso, R., Okamoto, T. & Okamoto, E. (2005), Practical Strong Designated Verifier Signature Schemes Based on Double Discrete Logarithms, *in* 'Information Security and Cryptology–CISC 2005', Vol. 3822 of *LNCS*, Springer, pp. 113–127.

Ustaoglu, B. (2008), 'Obtaining a secure and efficient key agreement protocol from (H)MQV and NAXOS', *Des. Codes Cryptography* **46**(3), 329–342.

A Preliminaries

In this section, we briefly describe the decisional Diffie-Hellman assumption, message authentication codes and pseudorandom functions.

Notations. We denote by $\mathbb{N}$ the set of natural numbers. An event is negligible in a security parameter k if it happens with a probability that is less than the inverse of any polynomial in k.

A.1 The DDH Assumption.

Let k be a security parameter and let $\mathbb{G}$ be a group of prime order q such that $|q| = k$ and let $g \in \mathbb{G}$ be a generator of G. Consider the following distributions:

$$\begin{aligned} \mathcal{DH}_{\mathbb{G}} &= \{(g, g^a, g^b, g^{ab})\ for\ a, b \xleftarrow{R} \mathbb{Z}_q\} \text{ and} \\ \mathcal{R}_{\mathbb{G}} &= \{(g, g^a, g^b, g^c)\ for\ a, b, c \xleftarrow{R} \mathbb{Z}_q\} \end{aligned}$$

We say that DDH assumption holds in $\mathbb{G}$ if $\mathcal{DH}_{\mathbb{G}}$ and $\mathcal{R}_{\mathbb{G}}$ are indistinguishable for all polynomial-time adversaries $\mathcal{A}^{ddh}$. More formally, the advantage of $\mathcal{A}^{ddh}$ given as $\mathsf{AdvDDH}_{\mathcal{A}^{ddh}}(k) = |\Pr[\mathcal{A}^{ddh}(1^k, \rho) = 1 | \rho \xleftarrow{R} \mathcal{DH}_{\mathbb{G}}] - \Pr[\mathcal{A}^{ddh}(1^k, \rho) = 1 | \rho \xleftarrow{R} \mathcal{R}_{\mathbb{G}}]|$ is negligible in k.

A.2 Pseudo-Random Function Family

Let $\mathsf{F} = \{f_s\}_{s \in S}$ be a family of functions for security parameter $k \in \mathbb{N}$ and with seed $s \in S = S(k) = S(k)$. Let $\mathcal{C}$ be an adversary that is given oracle access to either F_s for $s \in_R K$ or a truly random function with the same domain and range as the functions in F. F is said to be pseudorandom if $\mathcal{C}$'s advantage in distinguishing whether it has access to a random member of F or a truly random function is negligible in k, for all polynomial-time adversaries $\mathcal{C}$. That is $\mathsf{AdvPRF}_{\mathsf{F},\mathcal{C}}(k) = |\Pr[\mathcal{C}^{F_s(\cdot)}(1^k) = 1] - \Pr[\mathcal{C}^{Rand(\cdot)}(1^k) = 1]|$ is negligible in k.

A.3 Message Authentication Codes

A message authentication code (MAC) contains two algorithms; Mac and verify. The Mac algorithm takes a message m and a random key K as input and returns a tag τ. The verify algorithm takes m, K and the purported tag τ as input and returns *true* if τ is valid tag on m under K and *false* otherwise.

The standard notion of security considered for analyzing MAC schemes is strong existential unforgeability against chosen message attack (SUF-CMA) (Bellare, Kilian & Rogaway 1994). The SUF-CMA game is briefly described here.

The adversary $\mathcal{F}$ is given access to two oracles tag and verify whose behavior is as below:

tag: On a query to this oracle with an input message m, the challenger returns a tag τ computed on m using K.

verify: On a query to this oracle with input (m, τ), the challenger outputs *true* if τ is a valid tag on m under K. Otherwise it returns *false*.

$\mathcal{F}$ wins the game if it outputs a message-tag pair (m^*, τ^*) such that τ^* is a valid tag on m^* under K and τ^* was never returned from the tag oracle on input m^*. The advantage of $\mathcal{F}$, $\mathsf{AdvMac}_{\mathcal{F}}$ is its probability of success in winning the SUF-CMA game. We say that a MAC scheme $\mathcal{M}$ is SUF-CMA secure if $\mathsf{AdvMac}_{\mathcal{F}}$ is negligible in k.

A method for designing a MAC is using a PRF. The following theorem is proven in (Goldwasser & Bellare 1996-2008)

Theorem 5. *Let MAC* : $\mathsf{Keys}MAC \times \{0,1\}^d \rightarrow \{0,1\}^s$ *be a family of functions, and let* $q, t \geq 1$ *be integers. Then*

$$\mathsf{Adv}^{uf-cma}_{MAC}(t, q, dq) \leq \mathsf{Adv}^{prf}_{MAC}(t', q) + \frac{1}{2^s} \qquad (1)$$

where $t' = t + O(s + d)$

B Security Proof for the OPKE protocol

Note that a OPKE protocol can provide forward secrecy only for the initiator. Hence, in the test session only the initiator is allowed to be corrupted after issuing the test query. This initiator could be either owner of the test session or partner to the test session (if exists). We consider two cases while proving the security of our protocol. In the first case we prove the security of the protocol when the initiator in the test session is not corrupted, whereas the second case proves the security of the protocol when the initiator in the test session is allowed to be corrupted. Note that we require $\mathcal{A}^\pi$ to be passive in the second case, hence there exists a matching session to the test session.

Let n_s be the upper bound on the number of sessions invoked by $\mathcal{A}^\pi$. We prove the security of the protocol in a sequence of games. We denote by S_i the event that $b' = b$ in Game i.

Case 1: Sender in the test session is not corrupted. In this case the owner of the test session may or may not have a partner.

Game 0. This is the original attack game described in Section 5.1. By definition we have

$$Adv_{\mathcal{A}^\pi} \quad = \quad |2 \cdot \Pr[S_0] - 1| \qquad (2)$$

Game 1. This is similar to the previous game except that if two different sessions between a pair of users output exactly the same message, the game aborts.

$$|\Pr[S_1] - \Pr[S_0] \quad \leq \quad \frac{n_s}{q} \qquad (3)$$

Game 2. This is the same as the previous game except that a random value $t \overset{R}{\leftarrow} \{1, 2, \ldots, n_s\}$ is chosen. If $\mathcal{A}^\pi$ does not choose the t-th session as the test session, the game aborts and $\mathcal{A}^\pi$ outputs a random bit. Let Guess be the event that $\mathcal{A}^\pi$ chooses t-th session as the test session.

$$\begin{aligned} \Pr[S_2] &= \Pr[S_2|\mathsf{Guess}]\Pr[\mathsf{Guess}] + \\ &\quad \Pr[S_2|\neg\mathsf{Guess}]\Pr[\neg\mathsf{Guess}] \\ &= \Pr[S_1] \cdot \frac{1}{n_s} + \frac{1}{2} \cdot \left(1 - \frac{1}{n_s}\right) \qquad (4) \end{aligned}$$

Game 3. Game 3 is the same as Game 2 with the following differences. Let P_i and P_j be the peers of the t-th session chosen in Game 2 with P_i as the initiator and P_j as the responder. The static Diffie-Hellman key between P_i and P_j is replaced a random element $Z \overset{R}{\leftarrow} \mathbb{G}$. The only difference between Game 2 and Game 3 is whether Z is the Diffie-Hellman of the public keys of P_i and P_j. Hence these two games are indistinguishable under the DDH assumption.

$$|\Pr[S_3] - \Pr[S_2]| \quad \leq \quad \mathsf{AdvDDH}_{\mathcal{A}^{ddh}}(k) \qquad (5)$$

Game 4. This game is the same as the previous game except the game aborts if there has been a query of the form $\mathsf{send}(P_j, P_i, t, m, R)$ such that m contains a valid tag and the t-th session does not have P_i as the partner. This is essentially the existential unforgeability game played for the security of the MAC used in the protocol. Hence we have

$$|\Pr[S_4] - \Pr[S_3]| \quad \leq \quad \mathsf{AdvMac}_{\mathcal{F}}(k) \qquad (6)$$

Game 5. This game is the same as the previous game with the following difference: A query of the form $\mathsf{send}(P_i, P_j, t, I)$, is answered with an outgoing message $Y \| \tau_i$ where $\tau_i = \mathsf{Mac}_{k_{ij}}(P_i \| P_j \| Y)$ for $Y \overset{R}{\leftarrow} \mathbb{G}$. Since Y is chosen uniformly at random from $\mathbb{G}$, we have

$$\Pr[S_5] \quad = \quad \Pr[S_4] \qquad (7)$$

Game 6. This is the same as the previous game except that the session key here is substituted by a random element from $\mathbb{G}$. Similar to the argument in Game 3, the games Game 6 and Game 5 are indistinguishable under the DDH assumption.

$$|\Pr[S_6] - \Pr[S_5]| \quad \leq \quad \mathsf{AdvDDH}_{\mathcal{A}^{ddh}}(k) \tag{8}$$

$\Pr[S_6]$ is $\frac{1}{2}$ since the adversary has no information about the session key.

By combining the equations 2-8 we have

$$\begin{aligned} Adv_{\mathcal{A}^\pi} \quad \leq \quad & 2n_s(2 \cdot \mathsf{AdvDDH}_{\mathcal{A}^{ddh}}(k) + \mathsf{AdvMac}_{\mathcal{F}}(k)) \\ & + \frac{2n_s}{q} \end{aligned} \tag{9}$$

Case 2: Sender in the test session is corrupted. In this case, the test session would have a partner, as the adversary needs to be passive in the test session.

Game 0. Same as Game 0 of Case 1.

Game 1. Same as Game 1 of Case 1.

Game 2. This is the same as the previous game except that a random value $t \overset{R}{\leftarrow} \{1, 2, \ldots, n_s\}$ is chosen. If $\mathcal{A}^\pi$ does not choose the t-th session or its matching session as the test session, the game aborts and $\mathcal{A}^\pi$ outputs a random bit. Let Guess be the event that $\mathcal{A}^\pi$ chooses either the t-th session or its matching session as the test session.

$$\begin{aligned} \Pr[S_2] &= \Pr[S_2|\mathsf{Guess}]\Pr[\mathsf{Guess}] \\ &\quad + \Pr[S_2|\neg\mathsf{Guess}]\Pr[\neg\mathsf{Guess}] \\ &= \Pr[S_1] \cdot \frac{2}{n_s} + \frac{1}{2} \cdot \left(1 - \frac{2}{n_s}\right) \end{aligned} \tag{10}$$

Game 3. This game is the same as the previous game with the following difference. Let P_i and P_j be the peers of the t-th session chosen in Game 2 with P_i as the initiator and P_j as the responder. In this game the public key of P_j is replaced with an element $X \overset{R}{\leftarrow} \mathbb{G}$. This is game indistinguishable from the previous game as X is from the same distribution as the public key of P_j.

$$\Pr[S_3] \quad = \quad \Pr[S_2] \tag{11}$$

Game 4. This game is the same as the previous game with the following difference: A query of the form $\mathsf{send}(P_i, P_j, t, I)$, is answered with an outgoing message $Y \| \tau_i$ where $\tau_i = \mathsf{Mac}_{k_{ij}}(P_i \| P_j \| Y)$ for $Y \overset{R}{\leftarrow} \mathbb{G}$. Since Y is chosen uniformly at random from $\mathbb{G}$, we have

$$\Pr[S_4] \quad = \quad \Pr[S_3] \tag{12}$$

Game 5. This is the same as the previous game except that the session key here is substituted by a random element from $\mathbb{G}$. Game 4 and Game 5 are indistinguishable under the DDH assumption.

$$|\Pr[S_5] - \Pr[S_4]| \quad \leq \quad \mathsf{AdvDDH}_{\mathcal{A}^{ddh}}(k) \tag{13}$$

$\Pr[S_5]$ is $\frac{1}{2}$ since the adversary has no information about the session key.

By combining Equations 2,3,10,11,12,13 we have

$$Adv_{\mathcal{A}^\pi} \quad \leq \quad n_s \cdot \mathsf{AdvDDH}_{\mathcal{A}^{ddh}}(k) + \frac{2n_s}{q} \tag{14}$$

From Equations 9 and 14, the advantage of $\mathcal{A}^\pi$ in Case 1 is greater than that in Case 2. Hence we have $Adv_{\mathcal{A}^\pi} \leq 2n_s(2 \cdot \mathsf{AdvDDH}_{\mathcal{A}^{ddh}}(k) + \mathsf{AdvMac}_{\mathcal{F}}(k)) + \frac{2n_s}{q}$.

C Security proof for the SDV signature scheme

C.1 Proof of Theorem 2

Proof. We prove that the proposed SDV signature scheme is secure under the UF-CMA notion in a sequence of games. Let S_i the event that $\mathcal{A}^u$ outputs a valid SDV signature $(m^*, \sigma^*, pk_s^*, pk_v^*)$ in Game i as per the definition of UF-CMA in Section 3.1.

Game 0. This is the original UF-CMA game for the SDV signature signatures as defined in Section 3.1. All the queries of $\mathcal{A}^u$ are answered as per the definition. Hence we have

$$\mathsf{Adv}_{\mathcal{A}^u}(k) \quad = \quad \Pr[S_0] \tag{15}$$

Game 1. This is the same game as the previous one with the following difference. A signer U_s^* and a verifier U_v^* are chosen at random at the beginning of the game. If the adversary $\mathcal{A}^u$ corrupts any of the two chosen parties or outputs a forgery where U_s^* and U_v^* are not the signer and verifier, then the game aborts. U_s^* and U_r^* are therefore a guess at the parties that $\mathcal{A}^u$ will choose for the forgery out of the total of n parties. The probability that the guess is correct is $1/(n(n-1))$. Hence

$$\Pr[S_1] \leq \frac{1}{n(n-1)} \Pr[S_0] \tag{16}$$

Game 2. This is the same game as the previous one except that the sign and verify queries involving U_s^* as the signer and U_v^* as the verifier are modified as follows.

sign: on input $m \in \{0,1\}^*$, compute U and τ_1 as usual, but compute $\tau_2 = \mathcal{M}_\kappa(m)$ using a random key κ. If U has already been used in a sign query corresponding to U_s^* and U_r^*, then the same key κ is used.

verify: On input (U, τ_1, τ_2) on a message m, output valid if (U, τ_1) has been output before on a sign query corresponding to U_s^* and U_v^* and $\tau_2 = \mathcal{M}_\kappa(m)$ where κ is the one used previously in the sign query. Otherwise output invalid.

The only difference between Game 2 and Game 1 is whether a real session key or a random session key, which by Theorem 1 are indistinguishable, is used in producing the signature. Hence

$$|\Pr[S_2] - \Pr[S_1]| \leq \mathsf{Adv}_{\mathcal{A}^\pi}(k) \tag{17}$$

In Game 2, the event S_2 corresponds to $\mathcal{A}^u$ outputting a forgery m^*, τ_1^*, τ_2^* for signer U_s^* and verifier U_v^*, where U^*, τ_1^* have been output previously in a sign query . Since U^*, τ_1^* are independent of κ, $\mathcal{A}^u$ has no more information to produce a forgery corresponding to κ that an adversary against the MAC function $\mathcal{M}$. Hence

$$\Pr[S_2] \leq \mathsf{AdvMac}_{\mathcal{F}}(k) \tag{18}$$

From equations 15 to 18 we have

$$\mathsf{Adv}_{\mathcal{A}^u}(k) \leq n(n-1)\left(\mathsf{AdvMac}_{\mathcal{F}}(k) + \mathsf{Adv}_{\mathcal{A}^\pi}(k)\right) \tag{19}$$

□

C.2 Proof of Theorem 3

Proof. We prove that the proposed SDV signature scheme is secure under the IV-CMA notion in a sequence of games. Let S_i the event that $\mathcal{A}^i$ outputs a bit b' such that $b' = b$ in Game i as per the definition of IV-CMA in Section 3.2.

Game 0. This is the original IV-CMA game for the SDV signatures as defined in Section 3.2. All the queries of $\mathcal{A}^i$ are answered as per the definition. Hence we have

$$\mathsf{Adv}_{\mathcal{A}^i}(k) \quad = \quad |2 \cdot \Pr[S_0] - 1| \tag{20}$$

Game 1. This is the same game as the previous one except that the behavior of the challenger in this game is simulated by an adversary $\mathcal{A}^\pi$ against the OPKE protocol in Section 5.

sign: On input (m, pk_s, pk_v), $\mathcal{A}^\pi$ issues an establish-session$(pk_s, pk_v, \mathsf{sid}, I)$ and gets the outgoing message (U, τ_1). It obtains the computed session key through a reveal(U_s, sid) query and then uses the session key to compute a tag τ_2 on the message m using the MAC scheme. $\mathcal{A}^\pi$ finally returns (U, τ_1, τ_2) as the SDV signature to $\mathcal{A}^i$.

verify: On input (m, σ, pk_s, pk_v), $\mathcal{A}^\pi$ first parses the SDV signature σ as (U, τ_1, τ_2). It issues an establish-session$(pk_v, pk_s, \mathsf{sid}, (U, \tau_1), R)$ and once the session is accepted obtains the session key via reveal(U_v, sid) query. This key is then used to verify if τ_2 is a valid tag on the message m.

corrupt: If $\mathcal{A}^i$ issues a corrupt query with a user's identity U as input, $\mathcal{A}^\pi$ answers this query with the long-term private key of the user U by issuing a corrupt(U) query to its challenger.

When the event S_1 happens, $\mathcal{A}^\pi$ uses it to gain advantage against the OPKE protocol as follows:

At the end of stage 1, $\mathcal{A}^i$ outputs (m^*, pk_s^*, pk_v^*). $\mathcal{A}^\pi$ establishes t-th session by issuing an establish-session$(pk_s^*, pk_v^*, \mathsf{sid}^*, I)$ and obtains the outgoing message (U, τ_1) of the session. Now $\mathcal{A}^\pi$ can select the test session in one of the following ways.

1. the t-th session at U_s^* or
2. the matching session at U_v^* after issuing an establish-session$(pk_v^*, pk_s^*, \mathsf{sid}^*, \sigma_1^*, R)$

As a response to the test query, the challenger gives K_b to $\mathcal{A}^\pi$ as described in Section 5.1. $\mathcal{A}^\pi$ computes τ_2^* as a tag on the message m^* using the key K_b. The signature $\sigma^* = (U^*, \tau_1^*, \tau_2^*)$ is given to $\mathcal{A}^i$ as an output of the challenge phase.

Finally, when the event S_1 happens, $\mathcal{A}^\pi$ simply forwards the bit b' to its challenger. Thus we have

$$|\Pr[S_1] - \Pr[S_0]| \leq \mathsf{Adv}_{\mathcal{A}^\pi}(k) \tag{21}$$

Game 2. This is the same game as the previous one with the following difference: The MAC τ_2^* in the challenge SDV signature $(U^*, \tau_1^*, \tau_2^*)$ given to $\mathcal{A}^i$ is replaced by $\tau_r \xleftarrow{R} \{0,1\}^s$. By the pseudorandomness of the underlying PRF, from Equation 1 we have

$$|\Pr[S_2] - \Pr[S_1]| \leq \mathsf{AdvMac}_{\mathcal{F}}(k) - \frac{1}{2^s} \tag{22}$$

As the challenge signature in this game is uniformly distributed in the signature space, we have $\Pr[S_2] = 1/2$. From equations 20 to 22 we have

$$\mathsf{Adv}_{\mathcal{A}^i}(k) \leq 2(\mathsf{AdvMac}_{\mathsf{prf}}(k) + \mathsf{Adv}_{\mathcal{A}^\pi}(k) - \frac{1}{2^s})(23)$$

□

C.3 Proof of Theorem 4

Proof. We prove that the proposed SDV signature scheme is secure under the NT-CMA notion. Let S_0 the event that $\mathcal{A}^i$ outputs a bit b' such that $b' = b$ in Game 0 as per the definition of NT-CMA in Section 3.3.

The proof involves describing a polynomial algorithm simulate that can generate a signature that has the same distribution as an original SDV signature. simulate takes as input the public of the signer, the private key of the designated verifier and the message on which the simulated SDV has to be generated. The proof is straightforward once we describe the algorithm.

Game 0. This is the original NT-CMA game for the SDV signatures as defined in Section 3.3. The sign, verify and corrupt queries as normal. The simulate query is answered by the executing the following simulate algorithm.

simulate: On input (m, pk_s, pk_v), the challenger chooses $u \xleftarrow{R} \mathbb{Z}_q$ and computes $U = g^u$. It then computes the static Diffie-Hellman key between U_s and U_v using the private key U_v as $K_{sv} = pk_s^{sk_v}$. The session key is generated as $k_{sv} = U^{sk_v}$. The tags τ_1 and τ_2 are computed as $\tau_1 = \mathcal{M}_{K_{sv}}(U)$ and $\tau_2 = \mathcal{M}_{k_{sv}}(m)$. The simulated signature (m, τ_1, τ_2) is returned to $\mathcal{A}^n$.

As per the definition we have

$$\mathsf{Adv}_{\mathcal{A}^n}(k) \quad = \quad |2 \cdot \Pr[S_0] - 1| \tag{24}$$

Note that the output of the simulate algorithm has exactly the same probability distribution as the output of the sign algorithm. Moreover, the simulated signature is always verified to be valid by $\mathcal{A}^n$. Hence, we have $\Pr[S_0] = \frac{1}{2}$. Hence

$$\mathsf{Adv}_{\mathcal{A}^n}(k) \quad = \quad 0 \tag{25}$$

□