Strong designated verifier signature scheme: new definition and construction

Zuhua Shao
Zhejiang University of Science and Technology
No. 318, LiuHe Road, Hangzhou, Zhejiang, 310023, P. R. of China
zhshao_98@yahoo.com

Abstract: Recently, several strong designated verifier signature schemes have been proposed in the literature. In this paper, we first point out that such so-called strong designated verifier signature scheme is just message authentication code HMAC. Without the key property, unforgeability, for signatures, these schemes cannot enable signers to have complete controls over their signatures as demanded by Chaum and Van Antwerpen originally. No signer would use such Designated Verifier Signature schemes if he does not trust the designated verifier entirely. Then we introduce a new notion for the strong designated verifier signature scheme and its security requirements. We further propose the first strong designated verifier signature scheme based on Schnorr signature scheme, and provide a formal security proof under the DL assumption and the CDH assumption in the random oracle model. Finally, we discuss general methods to construct the strong designated verifier signature scheme.
Keyword: Undeniable signature; Designation of verifiers; Discrete logarithm; Random oracle model.

1 Introduction

Twenty years ago, Chaum and Van Antwerpen introduced a new primitive of cryptology, undeniable signature [4]. The motivation came from the following scenario: A software vendor puts digital signatures on its products to allow it to authenticate them as correct, free of viruses, etc, but only wants paying customers to be able to verify the validity of these signatures. Undeniable signatures can address this problem, which allows signers to have complete controls over their signatures. The verification of a valid undeniable signature requires the participation of the signer in an interactive confirmation protocol. On the other hand, the signer can prove that an alleged signature is a forgery through an interactive disavowal protocol. This property prevents the recipient from proliferating the copies of signatures without the signer's consent. Undeniable signatures find various applications in the real world such as licensing software, electronic cash, electronic voting and auctions.

However, this kind of signatures does not always achieve its goal, since this, in itself, only allows the prover to decide when a signature is verified and not by whom (or even by how many), because of blackmailing attacks [6] and mafia attacks [5].

In Eurocrypt'96, Jakobsson et al. [10] suggested a solution, designation of verifiers, that wanted to resolve the conflict between authenticity and privacy, and to dodge the described attacks by limiting who can be convinced by a proof. They showed that how, with only small change in the confirmation protocol for undeniable signatures, the confirmer could designate verifiers. Jakobsson et al. first introduced the concept of Designated Verifier Signature schemes (DVS),

They suggested that the DVS must satisfy the following requirement:
“Only the specified verifier can be convinced by the proof, even if he shares all his secret information with entities that want to get convinced” .
But this requirement is not necessary in the real world. The private key of verifier, as its name implies, can not be shared with other entities. A group of users sharing the same private key must be considered to be a user since they must hold the same accountability for the same private key in law. They further introduced the concept of Strong Designated Verifier Signature schemes (SDVS), in which no third party other than the designated verifier could even verify the validity of a designated verifier signature, since the designated verifier’s private key was required in verifying phase. In [18], Saeednia et al. firstly formalized the notion of the strong designated verifier signature, and proposed an efficient scheme. No signers would use such Designated Verifier Signature schemes those are harmful to themselves. Saeednia et al. put forward a new property, source hiding, the strong designated verifier signature scheme must satisfy. Given a message and its signature, it is infeasible to determine who from the original signer or the designated verifier generated the signature, even if one knows all the private keys. This property is accomplished by the designated verifier’s capability of creating another signature designated to him which is indistinguishable from the signer’s signature. Consequently, if there is some dispute between the original signer and the designated verifier, outsiders would have not any way to arbitrate it. It follows that the key property, unforgeability, for signatures, is no longer required.
Originally, the Designated Verifier Signature schemes result from the signer’s distrusts in the designated verifier. Now, the signer must bear common responsibility of the signatures generated by the designated verifier together with the designated verifier. It is the unhappiest outcome for the signer to use such so-called Designated Verifier Signature schemes. No signers would use such Designated Verifier Signature schemes those are harmful to themselves.
Recently, several strong designated verifier signature schemes and their identity-based variants have been proposed in the literature [16, 13, 14, 8, 12, 3, 23, 22, 11, 21, 20, 15]. Among them, the most efficient schemes have the verification equations [9]:

$$\sigma = H(m, k), \text{ where } k = y_A^{x_B} \bmod p = y_B^{x_A} \bmod p, \text{ or } k = \hat{e}(Q_{IDB}, S_{IDA}) = \hat{e}(S_{IDB}, Q_{IDA}).$$

As results, abandoning the essential requirement (unforgeability) for signatures, these so-called strong designated verifier “signature” schemes have been degenerated into the keying hash function [2]. Obviously, such simple message authentication code HMAC cannot accomplish the task proposed by Chaum and Van Antwerpen originally, i.e., completely controlling verification of signatures with non-repudiation.
Non-repudiation is the priority of any signature scheme. Sacrificing non-repudiation to exchange source hiding is more kicks than halfpence, “penny wise and pound foolish”
Therefore, the studies of so-called designated verifier "signatures", originating from Jakobsson, developed by Saeednia, followed by many researchers, have gone astray.

To achieve the goal putted forward by Chaum and Van Antwerpen, the strong designated verifier signature, just as its name implies, must satisfy two security requirements: unforgeability and unverifiability. Only the original signer can generate signatures and only the designated verifier can verify signatures. In this paper, we would like to provide new definition for the strong designated verifier signature and its security requirements. We will propose a concrete scheme

based on Schnoor signatures [19], and provide a formal security proof under the DL assumption and the CDH assumption in the random oracle model. Moreover we will discuss general methods to construct strong designated verifier signature scheme.

The rest of this paper is organized as follows. In Section 2, we propose the definition of strong designated verifier signature scheme and its security model. In section 3, we present a concrete strong designated verifier signature scheme based on Schnoor signature. We discuss the general methods to construct strong designated verifier signature scheme, which concludes this paper.

2 The definition of strong designated verifier signature scheme and its security model

In this section, we first present a precise definition of the strong designated verifier signature scheme, and then describe the security requirements by providing its security model.

2.1 The definition of the strong designated verifier signature scheme

Definition 1 (Strong designated verifier signature scheme)

A strong designated verifier signature scheme is a triple of algorithms (**KeyGen**, **Sign**, **Ver**):

- The key generation algorithm **KeyGen** that when given a security parameter 1^k as input, outputs two pairs (sk_j, pk_j) of matching private key and public key for the signer S and the designated verifier V respectively, $j \in \{S, V\}$. It is clear that **KeyGen** must be a probabilistic algorithm.
- The signing algorithm **Sign** that when given the (sk_S, pk_S, pk_V, m) of the matching private key sk_S and public key pk_S of the signer, the public key pk_V of designated verifier and a message m as input, produces a signature σ. The signing algorithm might be probabilistic, and in some schemes it might receive other input as well.
- The verification algorithm **Ver** that on input (sk_V, pk_S, m, σ), the private key sk_V of the designated verifier, the public key pk_S of the signer together with the message m and its signature σ, obtains either *invalid* or *valid*, with property that if $(sk_S, pk_S, sk_V, pk_V) \leftarrow$ **KeyGen**(1^k) and $\sigma \leftarrow$ **Sign**(sk_S, pk_S, pk_V, m), then **Ver**$(sk_V, pk_S, m, \sigma) = valid$. In general, the verification algorithm need not be probabilistic.

2.2 Security requirements and security models

The standard definition of the security of ordinary signature schemes, together with the first construction that satisfies it, was given by Goldwasser et al. [7]. Hence the strong designated verifier signature scheme must be existential unforgeable against adaptive chosen message attacks (EUF-CMA). However, the definition of ordinary EUF-CMA should be modified since the designated verifier might be an adversary. Additionally, the strong designated verifier signature scheme demands that only designated verifier can verify the validity of the signatures by using his private key.

Informally, there are two security requirements:

1. Unforgeability requires that nobody, including designated verifier, could forge any signature without the knowledge of the private key of the signer.
2. Unverifiability requires that nobody, except for signer, could verify any signature without the knowledge of the private key of the designated verifier.

Security model for unforgeability

We say that a strong designated verifier signature scheme is Strong UnForgeable against adaptive

Chosen Message Attacks (SUF-CMA) launched by a designated verifier adversary if no polynomial bounded adversary *A* has a non-negligible advantage against the challenger in the following game:

KeyGen: The challenger takes as input 1^k, runs the randomized algorithm to generate the key pair (sk_S, pk_S) of the signer and a public cryptographic hash function *H*. The adversary *A* chooses a group of key pairs (sk_{V0}, pk_{V0},…, sk_{Vn}, pk_{Vn}) of the designated verifier. The designated verifier's public keys should be certified by a certificate authority.
Queries: *A* issues signature queries adaptively on messages of his choice with respect to the public key pk_S of the signer and any public key pk_{Vi} of the designated verifier.
Response: Finally, the adversary *A* outputs a new designated verifier signature under the public keys pk_S of the signer and one public key pk_V of the designated verifier.
The adversary *A* wins the game if the output signature is nontrivial, i.e. it is not the answer of a signature query. The probability is over the coin tosses of the key generation algorithm and of *A*.

Remark 1**:** Here it is strong unforgeability [1], where the adversary needs to forge a new signature of a message and is allowed to ask for more signatures of the same message many times, and each new answer would give him some useful information. This more liberal rule makes the adversary successful when it outputs one new signature on a previously signed message.
Remark 2**:** This is a chosen key model, the adversary is allowed to choose the public keys of the designated verifier. As usually, theses public keys should be certified by a CA. The CA should verifies that an applicant knows that the corresponding private key by demanding the applicant to send the *CA* a signature, under the public key it is attempting to get certified, of some message that includes the public key and the identity of the applicant.

Security model for unverifiability
We say that a strong designated verifier signature scheme is UnVerifiable against adaptive Chosen Message Attacks (UV-CMA) if no polynomial bounded adversary *A* has a non-negligible advantage against the challenger in the following game:

KeyGen: The challenger takes as input 1^k, runs the randomized algorithm to generate the key pair (sk_S, pk_S) of the signer and the key pairs (sk_{V0}, pk_{V0},…, sk_{Vn}, pk_{Vn}) of a group of the designated verifiers and a public cryptographic hash function *H*. Finally, the challenger gives the public keys (pk_S, pk_{V0}, pk_{V1},…, pk_{Vn}) and the hash function *H* to the adversary *A*.
Queries: *A* issues queries q_1, ..., q_m adaptively, where query q_i is one of:
- Designated verifier signature query $<m_i, pk_{Vi}>$.
- Signature verify query $<m_i, \sigma_i, pk_{Vi}>$.

Challenge: *A* outputs two messages m_{i0}, m_{i1} as well as the two public keys pk_{Vi0}, pk_{Vi1} on which it wishes to be challenged.
The challenger picks a random bit $b \in \{0, 1\}$ and sets $\sigma_b = \text{Sign}(m_{ib}, pk_{Vib})$. It sends σ_b as the challenge to the adversary *A*.
The adversary *A* asks more signature queries and signature verify queries adaptively.
The only constraint is that σ_b did not appear in any signature verify query.
Guess: Finally, the adversary *A* outputs a guess $b' \in \{0, 1\}$ and wins the game if $b = b'$.

We define adversary *A*'s advantage in attacking the designated verifier signature scheme as the following function of the security parameter k (k is given as input to the challenger): $\mathrm{Adv}_A(k) = 2\Pr[b = b'] - 1$.
The probability is over the random bits used by the challenger and the adversary.

Remark 3: This security notion requires that it is computational infeasible to distinguish between two messages and two public keys, chosen by the adversary, which message has been signed with respect to which verifier's public key, with a probability significantly better than one half.

3 The concrete strong designated verifier signature

In this section, we present a concrete strong designated verifier signature and provide a formal security proof.

3.1 The concrete strong designated verifier signature

KeyGen: Take a security parameter 1^k as input, run a randomized key generation algorithm to generate a group $G_{g,p} = \{g^0, g^1, \ldots, g^{q-1} \bmod p\}$, where p and q are large primes and g is a generator of order q, and choose a public collision-free hash functions H: $\{0, 1\}^* \times Z_p^* \times Z_p^* \times Z_p^* \rightarrow Z_q^*$. The signer picks up at random x_S in Z_q^* as his private key and computes his public key $Y_S = g^{x_S} \bmod p$. The designated verifier generates his key pair $\{x_V, Y_V\}$ similarly.

Sign: For a message $m \in \{0, 1\}^*$, the key pair $\{x_S, Y_S\}$ of the signer and the pubic key Y_V of the designated verifier, first pick up at random t in Z_q^*, and compute $r = g^t \bmod p$, $k = y_V^{x_S} \bmod p$, $e = H(m, r, Y_V, k)$ and $s = t - ex_S \bmod q$. The strong designated verifier signature is $\sigma = (e, s)$.
Verify: Given a signature $\sigma = (e, s)$, the public key Y_S of the signer and the private key x_V of the designated verifier, check the verification equation $e = H(m, g^s y_S^e \bmod p, Y_V, y_S^{x_V} \bmod p)$.

Completeness: Because $s = t - ex_S \bmod q$, $r = g^t \bmod p$, implies $r = g^s y_S^e \bmod p$. Moreover, $y_S^{x_V} \bmod p = k = y_V^{x_S} \bmod p$. Hence, the signature $\sigma = (e, s)$ produced by the signing algorithm **Sign** is always *valid*.

Note that our results can also be carried over to other groups, such as those built on elliptic curves.

3.2 Security proof

Security of the proposed strong designated verifier signature is related to the following complexity assumptions:

3.2.1 Security assumptions

The **Discrete logarithm** problem : Given $(g, g^a \bmod p)$, compute a for any $a \in Z_p^*$.
Definition 2 (DL assumption)
A probabilistic algorithm D is said to (t, ε)-break a DL problem in the group $G_{g,p}$, if on input $(g, g^a \bmod p)$ and after running in at most t steps, D computes the discrete logarithm $\mathrm{DL}_g(g^a) = a$ with

probability at least ε, where the probability is taken over the coins of D and a chosen uniformly from Z_p^*.

We say that the (t, ε)-DL problem assumption holds in $G_{g,p}$ if no t-time algorithm has an advantage at least ε in solving DL problem in $G_{g,p}$.

The **Computational Diffie-Hellman** problem : Given $(g, g^a, g^b \bmod p)$, compute $g^{ab} \bmod p$ for any $a, b \in Z_p^*$.

Definition 3 (CDH assumption)

A probabilistic algorithm C is said to (t, ε)-break the CDH problem in the group $G_{g,p}$, if on input $(g, g^a, g^b \bmod p)$ and after running in at most t steps, C computes $g^{ab} \bmod p$, with probability at least ε, where the probability is taken over the coins of C and a, b chosen uniformly from Z_p^*.

We say that the (t, ε)-CDH problem assumption holds in $G_{g,p}$ if no t-time algorithm has an advantage at least ε in solving CDH problem in $G_{g,p}$.

3.2.2 Security proof for unforgeability

In fact, the proposed signature scheme is a variant of Schnorr signature scheme [19], by adding the public key of the designated verifier and the Diffie-Hellman key into the hash function. Because, Schnorr signature scheme is strong existential unforgeable against adaptive chosen message attacks (EUF-CMA) under the DL assumption [17], so is the strong designated verifier signature scheme proposed. Although designated verifier's public keys are chosen by the adversary, they should be certified by the CA. We can extract the corresponding private keys in the random oracle model from the knowledge proofs in certifying public keys. With this trivial modification to that of [17], we can show unforgeability. For the reason of brevity, we do not provide it in this paper.

3.2.3 Security proof for unverifiability

Theorem: Suppose that the (t', ε')-CDH problem assumption holds in $G_{g,p}$, then the strong designated verifier scheme is $(t, q_H, q_S, q_V, \varepsilon)$-secure against undesignated verifiers on $G_{g,p}$ for all t and ε in the random oracle model, where

$$t' \le t + (n + 1 + 2q_S)C_{exp}(G_{g,p})$$

$$\varepsilon' \ge \varepsilon/2 - q_S\,(q_H + q_S)/q - q_v/(q^2 - q_H - q_S).$$

Here $C_{exp}(G_{g,p})$ denotes the computation of a long exponentiation in the group $G_{g,p}$.

Proof: We use the random oracle model to show the security of the strong designated verifier signatures against an undesignated verifier. Assume that we are given an undesignated verifier UV that $(t, q_H, q_S, q_V, \varepsilon)$-breaks the scheme. That is, UV is a probabilistic polynomial time computer program which is supplied with a long public sequence of random bits, and can ask a polynomial number of questions to the random oracles H, DS, SV. We want to construct a "simulator" algorithm C, which takes $(p, q, g, g^a, g^b \bmod p)$ as input. Algorithm C simulates the signature scheme to the undesignated verifier UV. Algorithm C answers UV's hash function queries H, designated verifier signature queries DS and signature verify queries SV, and tries to translate UV's possible verifies into a solution to the Computational Diffie-Hellman problem $(g, g^a, g^b \bmod p)$.

The following is the attack game:

Algorithm C sets $g^a \bmod p$ to be the public key Y_S of the signer and ($g^b g^{b_0}$, $g^b g^{b_1}$ …, $g^b g^{b_n}$

mod p) to be the public keys (Y_{V0}, …, Y_{Vn}) of a group of designated verifiers, where (b_0, b_1, …, b_n) are numbers in Z_q^* chosen by Algorithm C. Finally, Algorithm C gives the public keys (Y_S; Y_{V0}, Y_{V1}, …, Y_{Vn}) to the undesignated verifier UV.

At any time, the adversary UV can query hash oracles H, sign oracle DS and verify oracle SV. To response to these queries, C maintains two lists of tuples for the hash oracles H and the sign oracle DS. We refer to the two lists as the H-list and the S-list, respectively. The contents of the two lists are "dynamic" during the attack game. Namely, when the game starts, they are initially empty, but at the end of the game, they record all pairs of queries/answers.

Answering H-oracle queries: If UV issues a hash oracle query (m_i, r_i, Y_i, k_i) where $1 \le i \le q_H$, C looks up the H-list to get the corresponding answer. If ((m_i, r_i, Y_i, k_i), e_i) exists in the H-list, C answers with e_i. Otherwise C generates e_i from Z_q* uniformly at random, answers with it, and adds ((m_i, r_i, Y_i, k_i), e_i) to the H-list.

Answering sign-oracle queries: If UV issues a sign query (m_i, Y_i) where $1 \le i \le q_S$, C generates e_i, s_i from Z_q* uniformly at random, computes $r_i = g^{s_i} Y_S^{e_i} \bmod p$. Then C looks for an item ((*, *, Y_i, k_i), *) in the H-list to find k_i. Otherwise, C generates k_i from Z_q* uniformly at random. If there exists an item ((m_i, r_i, Y_i, k_i), e_i') in the H-list with $e_i' \neq e_i$, C aborts and restarts simulation (the probability of this unfortunate coincidence occurring is at most $(q_H + q_S)/q$.). C answers with ((m_i, Y_i), e_i, s_i), and adds ((m_i, r_i, Y_i, k_i), e_i) to the H-list, ((m_i, Y_i), e_i, s_i) to the S-list.

Answering verify-oracle queries: If UV issues a verify query (((m_i, Y_i), e_i, s_i), where $1 \le i \le q_V$, C looks for such item in the S-list. If C finds it, C answers with *valid*. Else if there exists an item ((m_i, r_i, Y_i, k_i), e_i) in the H-list, C answers with *invalid*. Otherwise, C aborts and restarts simulation. The probability that ((m_i, Y_i), e_i, s_i) is a valid signature and there is no item ((m_i, *, Y_i, *), e_i) in the H-list is at most $1/(q^2 - q_H - q_S)$.

According to the security model, UV outputs two sign queries. Without loss of generality, we assume that they are (m_0, Y_{V0}) and (m_1, Y_{V1}). C picks a random bit $b \in \{0, 1\}$ and sets σ_b = Sign(m_b, Y_{Vb}). It sends σ_b as the challenge to the adversary UV.

The adversary UV is allowed to continue to ask more signature queries and signature verify queries adaptively.

The only constraint is that σ_b did not appear in any signature verify query.

Finally, the adversary UV outputs a guess $b' \in \{0, 1\}$ and wins the game if $b = b'$.

By definition, $\Pr[b = b'] = \mathrm{Adv}_{UV}(k)/2 + 1/2$.

If neither (m_0, r_0, Y_{V0}, k_0) nor (m_1, r_1, Y_{V1}, k_1) has queried by UV to the *H-oracle* in the game, UV has no information about b since the answer to the hash function is randomly chosen, thus

$$\Pr[b = b'] = 1/2.$$

Suppose that (m_0, r_0, Y_{V0}, k_0) has queried in a *H-oracle query*, thus, $k_0 = y_S^{x_{V0}} = y_{V0}^{x_S} = g^{a(b+b_0)} \bmod p$ implies $g^{ab} = k_0/(g^a)^{b_0} \bmod p$. Algorithm C can find it in the H-list.

Therefore, $\varepsilon' \ge \varepsilon/2 - q_S(q_H + q_S)/q - q_V/(q^2 - q_H - q_S)$.

Similarly $t' \leq (t + (n + 1+2q_S)C_{exp}(G_{g,p}))$.
Q.E.D.

Remark : This proof only show that nobody could verify any signature without the knowledge of the Diffie-Hellman key between the signer and the designated verifier. Hence, both the signer and the designated verifier are able to verify the designated verifier signature. This is reasonable since the signer should certainly control the signatures he has signed.
On the other hand, given the Diffie-Hellman key, judges can resolve possible disputes between the signer and the designated verifier.

4 Conclusions

Because Jakobsson et al. gave up the non-repudiation requirement that is essential in the ordinary signature scheme, so-called strong designated verifier "signature" schemes have been degenerated into the keying hash function. Obviously, such simple message authentication code HMAC cannot accomplish the task proposed by Chaum and Van Antwerpen originally, i.e., completely controlling verification of undeniable signatures with non-repudiation. Moreover, the designated verifier is able to create another signature designated to him which is indistinguishable from the signer's signature. The signer must bear common responsibility of the signatures generated by the designated verifier together with the designated verifier. No signer would use such Designated Verifier Signature schemes if he does not trust the designated verifier entirely. Therefore, the research lines originated from Jakobsson et al. is not practical at all.
In this paper, we have solved this task, by introducing the new notion of strong designated verifier signature scheme. We have provided a new definition and new security requirements: unforgeability and unverifiability. Then we have proposed the first strong designated verifier "signature" scheme based on Schnoor signatures and have provided a formal security proof.
From the concrete scheme, we obtain a general method to construct a provably secure strong designated verifier signature scheme. Adding the Diffie–Hellman key to the hash function of a variety of signature scheme since the hash function is indispensable in any provably secure signature scheme.

Acknowledgements

The author would like to thank Ms. Xiaolan Zhang in the library of ZUST for her assistance in gathering material and information.

References

1. J. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption, in *Advances in Cryptology* (*Eurocrypt'02*), LNCS 2332, Springer-Verlag, Berlin, pp.83-107, 2002.
2. M. Bellare, R. Canetti, and H. Krawczyk, Keying hash functions for message authentication, in *Advanced in Cryptology* (Crypto'96), LNCS 1109, pp.1-15, Springer-Verlag, 1996.
3. J. F. Cao and Z. Cao , An identity based universal designated verifier signature scheme secure in the standard model, *The Journal of Systems & Software*, 82 (4), pp.643-649, Apr 2009.
4. D. Chaum and H. van Antwerpen, Undeniable signatures, in *Advances in Cryptology* (Crypto'89), LNCS 435, pp. 212-216, Springer-Verlag, 1990.
5. Y. Desmedt, C. Goutier, and S. Bengio, Special uses and abuses of the Fiat-Shamir passport protocol, in *Advances in Cryptology* (Crypto'87), LNCS 293, pp. 21-39, Springer-Verlag,

1998.

6. Y. Desmedt and M. Yung, Weaknesses of undeniable signature schemes, in *Advances in Cryptology* (Eurocrypt'91), LNCS 547, pp. 205-220, Springer-Verlag, 1991.
7. S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks, *SIAM Journal on Computing*, 17(2), pp.281-308, 1988.
8. X. Huang, Y. Mu, W. Susilo, and F. Zhang, Short designated verifier proxy signature from pairings, in *The First International Workshop on Security in Ubiquitous Computing Systems* (SecUbiq'05), LNCS 3823, pp. 835-844, Springer Verlag, 2005.
9. X. Huang, W. Susilo, Y.Mu and F. Zhang, Short designated verifier signature scheme and its identity-based variant, *International Journal of Network Security*, 6(1), pp. 82-93, 2008.
10. M. Jakobsson, K. Sako, and R. Impagliazzo, Designated verifier proofs and their applications, in *Advances in Cryptology* (Eurocrypt'96), LNCS 1070, pp.143-154, Springer-Verlag, 1996.
11. B. Kang, C. Boyd, and E. Dawson, Identity-based strong designated verifier signature schemes: Attacks and new construction, *Computers and Electrical Engineering*, 35(1), pp.49-53, Jan 2009.
12. B. Kang, C. Boyd, and E. Dawson, A novel identity-based strong designated verifier signature scheme, *The Journal of Systems & Software*, 82 (2), pp.270-273, Feb 2009.
13. F. Laguillaumie and D. Vergnaud, Designated verifiers Signature: anonymity and efficient construction from any bilinear map, in *Fourth Conference on Security in Communication Networks*'04 (SCN'04), LNCS 3352, pp. 107-121, Springer-Verlag, 2004.
14. F. Laguillaumie and D. Vergnaud, Multi-designated verifiers signatures, in *Information and Communications Security* (ICICS/04), LNCS 3269, pp. 495-507, Springer-Verlag, 2004.
15. J.-S. Lee and J. H. Chang, Comment on Saeednia et al.'s strong designated verifier signature scheme, *Computer Standards & Interfaces*, 31(1), pp.258-260, Jan. 2009.
16. H. Lipmaa, G. Wang and F. Bao, Designated Verifier Signature Schemes: Attacks, New Security Notions and A New Construction, in *The 32nd International Colloquium on Automata, Languages and Programming*, ICALP 2005, LNCS 3580, pp. 459-471, Springer-Verlag, 2004.
17. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures, *Journal of Cryptology*, 13(3), pp.361-396, 2000.
18. S. Saeednia, S. Kramer, and O. Markovitch, An efficient strong designated verifier signature scheme, in *The 6th International Conference on Information Security and Cryptology* (ICISC'03), pp. 40-54, Springer-Verlag, 2003.
19. C. P. Schnorr. Efficient signature generation by smart cards, *Journal of Cryptology*, 3(3), pp. 161-174, 1991.
20. S. H. Seo, J.Y. Hwang, K.Y. Choi, and D.H. Lee, Identity-based universal designated multi-verifiers signature schemes, *Computer Standards & Interfaces*, 30 (5), pp.288-295, Jul 2008.
21. K.A. Shim, Rogue-key attacks on the multi-designated verifiers signature scheme, *Information Processing Letters*, 107 (2), p.83-86, Jul 2008
22. Y. Yu, C. Xu, X. Zhang, and Y. Liao, Designated verifier proxy signature scheme without random oracles, *Computers and Mathematics with Applications*, 57(8), pp.1352-1364, Apr 2009.
23. J. Zhang and J. Mao, A novel ID-based designated verifier signature scheme, *Information*

Sciences, 178 (3), pp.766-773, Feb 2008.