Designated Verifier Signature: Definition, Framework and New Constructions

Yong Li[1], Willy Susilo[2], Yi Mu[2], and Dingyi Pei[1,3]

[1] State Key Laboratory of Information Security
Graduate School of Chinese Academy of Sciences,
Beijing 100049, P.R. China
li.yong9@gmail.com
[2] School of Information Technology and Computer Science
University of Wollongong, Wollongong NSW 2522, Australia
{wsusilo,ymu}@uow.edu.au
[3] Guangzhou University, Guangzhou 510400, P.R. China

Abstract. To date, there are numerous variants of designated verifier signatures (DVS), including the notion of strong DVS, multi DVS, universal DVS, etc. In this paper, *for the first time*, we present a *generic definition* of DVS model. We also explore the related security notions in DVS, including unforgeability, non-transferability and non-delegatability, and study the relationship of these notions against variants of DVS. Furthermore, we classify the multi designated verifier signature schemes into four categories depending on the way the verification and simulation is performed. We also point out some drawbacks on the existing DVS schemes, and finally present a new and efficient *constant size* multi DVS scheme that produces a constant size signature regardless the size of the receivers' group. Our scheme is proven secure in the standard model.

1 Introduction

Digital signatures, one of fundamental authentication methods, allow a signer who is equipped with a secret key to sign messages such that anyone can verify the authenticity of the messages with respect to the published public key. The publicly verifiable or self-authenticating property of conventional digital signature is not necessarily desirable in some real scenarios when user's privacy must be taken into account. For example, sometimes it is desirable that a verifier should not present the signatures to other parties, such as certificates for personal health records, income summary, negotiatory price between bidder and tenderee in e-auction system, etc. Designated verifier signature schemes (or DVS schemes, for short) are mainly constructed to address such signer privacy issue by *preventing* the signature from being arbitrarily disseminated [6]. Only a nominated party, called the designated verifier, will trust the authenticity of the message signed via DVS schemes. The intuition behind these schemes is to stop the verifier from transferring his conviction about validity of the signature to any third party, using the fact that the verifier himself can generate such a signature that is indistinguishable from the real signature generated by the signer. Thus, DVS do not provide non-repudiation property of traditional digital signatures.

To date, a number of DVS schemes and their variants have been proposed in the literature, such as universal DVS (UDVS) [12,18,17,5], strong DVS (SDVS) [13,15,16],

J. Indulska et al. (Eds.): UIC 2007, LNCS 4611, pp. 1191–1200, 2007.

multi-DVS (MDVS) [9,11,3], etc. Unfortunately, these abundance have distinct characteristics and addressing different security notions and hence, it has deviated from its original goal and motivation. Therefore, we are motivated to investigate these notions closely and classify them accordingly, so that the resulting framework can be used to analyze the security of DVS schemes and their variants.

Our Contributions. In this paper, we provide a generic definition of DVS model that captures its three variants, i.e. strong DVS, multi DVS and universal DVS, together with its security properties. Based on our framework, we classify the existing DVS variants to enable us to study the characteristics of these schemes.

On non-delegatability notion, we show that one of Zhang et al.'s ID-based UDVS schemes is delegatable. Our new analysis provides an alternative non-transferability proof for the scheme, although such delegation "attack" has been addressed recently in Susilo et al. [14].

On multi-designated verifier signature (MDVS), we analyze Chow's ID-based strong MDVS scheme [3] and show its security problem. The scheme is *fragile* under a forgery attack if the verifier behaves as described in their verification algorithm. Furthermore, we refine the classification of MDVS models. Specifically, there are four types of MDVS schemes according to the number of verifiers participating in the verification and simulation process. Subsequently, we point out that the three MDVS models [9,11,3] actually have subtle differences. Particularly, the two MDVS models [11,3] are different from the first MDVS model in [9] in the sense that the verification and simulation algorithms can *only* be performed by the *coalition* of *all* designated verifiers. Meanwhile, in the model [9], the verification can be performed by every verifier *independently* and simulation can be performed by *coalition* of all designated verifiers.

Finally, we present a new and novel construction of constant-size MDVS scheme which signature length does not grow linearly with the size of the verifiers. Furthermore, it's the first scheme that consistent with type of MDVS [9] and secure in the standard model.

Roadmap. The rest of this paper is organized as follows. In Section 2, we provide formal definition of DVS and its variants. In Section 3, we present the non-delegatability analysis on Zhang et al.'s ID-based UDVS scheme. In Section 4, we present the refined classification of multi DVS schemes, together with presenting the security weakness of Chow's ID-based strong MDVS scheme. In Section 5, we construct a new and efficient constant-size MDVS scheme. Section 6 concludes the paper.

2 DVS and Its Variants

In this section, we present a generic definition of designated verifier signature (DVS) and their variants that could be derived from it. Let S be the signer, and D be the designated verifier. Let $\mathcal{M}$ be the message space. A *designated verifier signature* (DVS) scheme is defined by the following algorithms:

- $\mathsf{Setup}(k)$ is a probabilistic algorithm that outputs the public parameter *param*, where k is the security parameter.
- $\mathsf{KeyGen}(param, k)$ is a probabilistic algorithm that takes the public parameters as an input and outputs a secret/public key-pair $(\mathsf{SK}, \mathsf{PK})$;

– $\mathsf{Sign}_{\mathsf{SK}_S,\mathsf{PK}_D}(m)$ takes as input signer's secret key, designated verifiers' public key, a message $m \in \mathcal{M}$ and a possible random string, and outputs a signature σ;
– $\mathsf{Verify}_{\mathsf{PK}_S,\mathsf{PK}_D}(m,\sigma)$ is a deterministic algorithm that takes as input a signing public key PK_S, public key of designated verifier D, a message $m \in \mathcal{M}$ and a candidate signature σ, and returns accept or reject;

We say that a signature σ on m is valid if $\mathsf{Verify}_{\mathsf{PK}_S,\mathsf{PK}_D}(m,\sigma) = \mathsf{accept}$. As usually, we require that an DVS scheme is correct, that is, for all $(\mathsf{SK}_S, \mathsf{PK}_S)$ and $(\mathsf{SK}_D, \mathsf{PK}_D)$ output by KeyGen, for $m \in \mathcal{M}$, we have

$$\mathsf{Verify}_{\mathsf{PK}_S,\mathsf{PK}_D}(\mathsf{Sign}_{\mathsf{SK}_S,\mathsf{PK}_D}(m)) = \mathsf{accept}.$$

Generally, a secure DVS scheme must satisfy the following two basic properties:

- ***unforgeability***, which is consistent with classical security notion for signature, namely, existential unforgeable against adaptive chosen message attack (EUF-CMA) [4]. More formally, it is defined using the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$:
  - Let $\mathcal{A}$ be the EUF-CMA adversary. In the startup of the game, $\mathcal{C}$ provides the common scheme parameter cp to $\mathcal{A}$, where $cp \leftarrow \mathsf{Setup}(k)$ and k is the security parameter.
  - $\mathcal{C}$ provides the signer's public key PK_S and verifiers' public key PK_D to $\mathcal{A}$.
  - At any time, $\mathcal{A}$ can query the signing oracle for the signature on any message m_i of his choice up to q_S times (which is polynomial in k). $\mathcal{C}$ will answer $\mathcal{A}$'s queries by providing the value $\sigma = \mathsf{Sign}(cp, \mathsf{SK}_S, m_i, \mathsf{PK}_D)$ where SK_S is the corresponding secret key of the specified user queried by $\mathcal{A}$ for m_i.
  - Eventually, $\mathcal{A}$ will output a new DVS σ^* for a message m^*. $\mathcal{A}$ wins the game if $\mathsf{Verify}_{\mathsf{PK}_S,\mathsf{PK}_D}(m^*,\sigma^*) = \mathsf{accept}$, and m^* has never been queried to the signing oracle before, for the designated verifier with public key PK_D.

  The success probability of an adversary to win the game is defined by $Succ^{EUF-CMA}_{DVS,\mathcal{A}}(k)$.

  Definition 1. *(Unforgeability) We say that a DVS scheme is existentially unforgeable under a chosen message attack, or (t,ε)-EUF-CMA secure, if no polynomially bounded adversary $\mathcal{A}$ running in time t has a success probability $Succ^{EUF-CMA}_{DVS,\mathcal{A}}(k) \geq \varepsilon$.*

- ***non-transferability.*** Formally, non-transferability is defined through the following game involving $\mathcal{D}$ and $\mathcal{C}$. Challenger $\mathcal{C}$ can simulate attack environment for $\mathcal{D}$ by running Sign and $\mathsf{Simulation}$ algorithm. $\mathcal{D}$ is a distinguisher that tries to distinguish whether a given output is generated by signer or designated verifier.
  - The challenger $\mathcal{C}$ takes as input a security parameter k and executes $cp \leftarrow \mathsf{Setup}(k)$, $(\mathsf{SK}_S, \mathsf{PK}_S) \leftarrow \mathsf{KeyGen}(k, cp)$, $(\mathsf{SK}_D, \mathsf{PK}_D) \leftarrow \mathsf{KeyGen}(k, cp)$. $\mathcal{C}$ provides $\mathsf{PK}_S, \mathsf{PK}_D$ to $\mathcal{D}$ and keeps $\mathsf{SK}_S, \mathsf{SK}_D$ secret.
  - The distinguisher $\mathcal{D}$ issues signing queries on any message m_i. $\mathcal{C}$ responds with $\sigma = \mathsf{Sign}(cp, \mathsf{SK}_S, m_i, \mathsf{PK}_D)$.

- $\mathcal{D}$ submits new message m^* to $\mathcal{C}$. $\mathcal{C}$ then flips a fair coin $b \overset{R}{\leftarrow} \{0,1\}$, generates a signature σ^* and returns it to $\mathcal{D}$. For example, if b=0, $\mathcal{C}$ runs Sign algorithm and returns $\sigma^* = \mathsf{Sign}(cp, \mathsf{SK}_S, m_i, \mathsf{PK}_D)$; Otherwise, $\mathcal{C}$ runs Simulation and returns $\sigma^* = \mathsf{Simulation}(cp, \mathsf{SK}_D, m_i, \mathsf{PK}_D)$.
- On receiving the challenging signature, $\mathcal{D}$ can issues new queries with the restriction of not querying m^*.
- Eventually, $\mathcal{D}$ outputs a bit b' and wins if $b' = b$.

The advantage of an adaptively chosen message distinguisher $\mathcal{D}$ is defined as $Adv\mathcal{D}^{CMA}_{DVS} = |Pr[b' = b] - 1/2|$.

Definition 2. *(Non-transferability) We say that a DVS scheme is non-transferability against a (t, q_S) adaptively chosen message distinguisher $\mathcal{D}$ if $Adv\mathcal{D}^{CMA}_{DVS}$ is negligible after making at most q_S signing queries in time t.*

2.1 Differences from Former Definitions

There are several differences between the new definition above and the previously adopted definitions [13,8,10,7]. The new definition is more versatile and extendable because other DVS variants could be derived from it easily. The three main variations of DVS in the literature were unfolded as follows.

A DVS scheme is called to be

- *strong* DVS (SDVS), if the verification algorithm also takes SK_D as an input. Moreover, the verification without SK_D is computationally infeasible.
- *Multi-* DVS (MDVS), if the signature is intended for n verifiers, $n > 1$.
- *Universal* DVS (UDVS), if it contains a conventional signing algorithm (w.r.t. no designated verifier) and an arbitrary signature holder (designator) can convert the conventional signature to a designated signature w.r.t. an arbitrary designated verifier.

We note that we do not further differentiate the DVS variants that are public key based from the identity-based setting. Our view is to capture the variants that are distinct and not only in the view of public key setting.

The three DVS variants were proposed respectively from a different angle. Multi-DVS can be treated as an extended DVS with multiple designated verifiers. Universal DVS allows multiple signature designators (from a signer to any signature holder). The strong DVS notion was proposed to enhance the DVS security for a more hostile environment [6,13,8]. Generally, to make a strong designated verifier signature, one will use the public key of the intended verifier to encrypt the signature, and this implies that only the specific verifier can verify the signature by using his/her private key.

Another difference is that we treat the simulation function (which mainly guarantees non-transferability) as one security requisites instead of as a component of signature algorithms like in [7]. Commonly, non-transferability is guaranteed by a signature *simulation* algorithm that is run by the designated verifier to produce an identically distributed signature that is indistinguishable from the original signature. Such a particular property differentiates DVS from conventional digital signatures. Finally, we treat another security notion of DVS, non-delegatability [10,7,14], as an *enhanced* security option.

3 Delegatability Analysis of ID-Based UDVS Scheme

In this section, we firstly analyze delegatability property on one of Zhang et al.'s ID-based UDVS schemes [19], according to Lipmaa et al.'s original non-delegatability definition in [10]. Furthermore, we interpret the delegatability analysis from new angle of view, inspired by the new refined non-delegatability notion in [14]. The brief review of ZSMC05 scheme [19] is omitted due to page limitation.

3.1 Delegatability Analysis

(Negative aspect) If signer leaks $d = e(SK_S, PK_D)$ or verifier leaks $d = e(PK_S, SK_D)$ to third party T, then T can compute the simulated designated verifier signature as follows.

- Compute $U = rPK_S$, where $r \in_R Z_q^*$, $h = H_1(U||m)$.
- Compute $\sigma' = d^{(r+h)}$.
- Output the designated verifier signature on m as (U, σ').

Clearly,

$$\begin{aligned} e(U + H_1(U||m)PK_S, SK_D) &= e(rPK_S + H_1(U||m)PK_S, SK_D) \\ &= e(PK_S, SK_D)^{r+H_1(U||m)} = e(SK_S, PK_D)^{r+h} \\ &= d^{(r+h)} = \sigma'. \end{aligned}$$

Thus, designated verification equation is satisfied. It means that the scheme has delegatability weakness according to Lipmaa et al.'s definition in [10]. The above "delegation attack" allows us to delegate the signing rights of a fixed signer with respect to a *fixed designated verifier*.

(Positive aspect) The above scheme is indeed a strong DVS scheme and the $d = e(SK_S, PK_D) = e(PK_S, SK_D)$ is a crucial *common key* in the scheme, with which anyone can perform simulation algorithm. According to the new refined non-delegatability notion in strong DVS [14], such an essential key should not be released. This makes the above delegatability analysis no longer significant.

Note that in proof for non-transferability ([19], Theorem 2), the simulation algorithm is actually the *same* as the designated verification algorithm. Interestingly, the computations in delegatability analysis could serve as the simulation algorithm for this scheme if verifier firstly uses his/her secret key to compute $d = e(PK_S, SK_D)$, which provides an alternative non-transferability proof for the scheme. That is, the analysis makes the scheme have different simulation and designated verification algorithm.

4 MDVS Model

To date, to the best of our knowledge, there is only a few MDVS schemes proposed in the literature (i.e. [9,11,3]). Interestingly, we found there is subtle distinction between the above three MDVS models, which has not been pointed out explicitly in the literature. Meanwhile, we also analyze some potential vulnerability of Chow's MDVS scheme [3], that will be presented in the next section.

4.1 Analysis of Chow's MDVS Scheme [3]

Due to page limitation, the brief review of Chow's scheme [3] is omitted.

Vulnerability Analysis
Suppose that an attacker intercepts a signature $\{U_1, U_2, V, Y, Z_1, Z_2, \cdots, Z_n\}$, and he tampers it with $\sigma' = \{U_1, U_2, V, Y, *, *, \cdots, Z_i, \cdots, *\}$. That is, he fills the fragment $\{Z_1, Z_2, \cdots, Z_n\}$ with random values except Z_i. We note that the forged signature σ' still satisfies V_i's verification equations. In fact, verifier V_i only checks the validity of the signature segment $\{U_1, U_2, V, Y, Z_i\}$ in the Verify algorithm.

However, other verifiers $V_j (j \neq i)$ could detect the invalidity of σ'. Unfortunately, since there is no correspondence or cooperation between n verifiers in their MDVS model, V_i will be cheated in this situation, by believing that the signature is indeed valid. Furthermore, the direct countermeasure that all verifiers cooperatively perform Verify algorithm will invalidate the above attack. Unfortunately, $4n$ costly pairing operations are required to perform this operation.

In short, if only one V_i participates in the verification (as the author claimed in [3] that there are only 4 pairing operations in Verify algorithm), the scheme is *fragile* under the above forgery attack. If the scheme is being adjusted to the case where every verifier cooperatively performs the verification steps, then it becomes very inefficient, and hence, we have a paradox.

Amendment of Chow's scheme. We present an alternative scheme to avoid the above attack. The Setup, Extract, Sign and Step 1,2 in Verify algorithm is the same as original scheme. Step 3 in the Verify algorithm is revised as follows.

Return true if the following equations

1. $e(P_{pub}, U_1 + h_1 H_1(ID_S) + U_2 + h_2 P_V) = e(P, V)$;
2. For each $Z_j (j \neq i)$, $e(Z_j, P) = e(H_1(ID_{V_j}) + Q, Y)$.

hold and $\perp$ otherwise.

This improvement fixes the flaw in Chow's scheme. There is totally $4 + 2 * (n - 1) = 2 * (n + 1)$ paring operations in the Verify algorithm. Moreover, the revised Verify algorithm could be performed by every verifier independently.

4.2 Refined Classification

In this section, we present a refined definition of MDVS scheme.

According to number of verifiers required in the Verify algorithm, there are two types of MDVS schemes:

(a) the Verify algorithm can be performed by every verifier *independently*;
(b) the Verify algorithm can be performed only by the *coalition* of all designated verifiers.

Analogously, there are other two kinds of MDVS if the number of verifiers in the signature simulation algorithm is taken into account.

(c) the simulation can be performed by every verifier *independently*;
(d) the simulation can be performed only by the *coalition* of all designated verifiers.

Table 1. Classification of MDVS schemes

Verify\Simulation	(c)	(d)
(a)	n/a	[9]
(b)	n/a	[11] [3]

Therefore, by combining the above categories, we classify four kinds of MDVS schemes. Furthermore, we classify the three existing MDVS schemes [9,11,3] using this classification as shown in Table 1. (The detail description of schemes refer to [9,11,3].)

We stress that the functionality of the simulation algorithm which guarantees non-transferability is *inherent* in DVS schemes. Unfortunately, it was not explicitly stated as a necessary ingredient in [9,3]. (In [11], such simulation capability of verifier was demonstrated in the proof of non-transferability.) According to Li, Lipmaa and Pei's analysis in [7], it needs all verifiers to work cooperatively to accomplish the simulation in [9]. The scheme in [3] also does not provide any simulation algorithm to ensure the non-transferability property. However, the construction is consistent with the generic MDVS construction mechanism from ring signatures from [9] and the author indicated that *signer ambiguity* is directly obtained from the underlying ID-based ring signature.

In conclusion, based on the above analysis and the table, the three MDVS models (all are strong MDVS) have subtle differences. Particularly, the subsequent two MDVS models [11,3] are different from the first MDVS model [9], with respect to the number of verifiers participating in the verification and simulation algorithms.

4.3 Transfer Mode in MDVS

The MDVS model is treated as a generalized DVS in a multi-user setting, in which the signature is intended to a specific set of different verifiers. In such a scenario, there is another question that should be considered in implementing practical MDVS scheme: "How does the signature propagate to all the verifiers?" We refer the signature propagation as the *signature transfer* in DVS schemes.

Essentially, there are two signature transfer modes, namely *sequential mode* and *broadcast mode*. In sequential mode, the signer delivers her signature from verifier V_1 to V_n in a step-by-step manner, while in broadcast mode, the signer broadcasts the signature to all verifiers simultaneously.

In sequential mode, if each verifier could simulate the signature independently, then successor V_j *cannot* distinguish whether σ was generated by the signer or simulated by his/her predecessor V_i ($i < j$). Such case *breaches* the basic requirement of MDVS and actually it degenerates into DVS, namely, only one specific recipient V_1 can be convinced that "signer has signed on a message". However, if the simulation must be performed by every verifier cooperatively, then V_j can identify the source of signature since he knows that he does not participate in the simulation algorithm.

In broadcast mode, supposing the simulation must be done by all verifiers collaboratively, if a verifier V_i is corrupted or captured/controlled by a hostile third party and hence becomes unavailable, then the simulation cannot be accomplished. In this case, all parties, including the designated verifier and any other third party, can be convinced with the authenticity of the signature, which deviates the purpose of having a designated verifier scheme. Hence, the non-transferability property is no longer satisfied by the scheme. We note that in a normal situation, non-transferability is always guaranteed by the signature simulation algorithm. Additionally, if each verifier can simulate the signature independently, then this case will not influence the characteristic of MDVS schemes.

According to the refined classification in section 4.2 and the above analysis, the secure implemental approach for [9,11,3] is to adopt sequential mode when transferring signature to multiple intended recipients. Furthermore, in [11,3] schemes, every verifier need to be well protected to guarantee their availability, otherwise, no verifier could check the validity of the signature. Simultaneously, to the best of our knowledge, there is no MDVS scheme adapted for broadcast mode, of which the simulation algorithm could be independently executed by each verifier. We leave it as an open problem to explore such scheme.

Therefore, the signature transfer mode is an important issue in implementing a practical MDVS scheme. Unfortunately, the distinction on verification/simulation mode (independently or cooperatively) and correlative problem of signature transfer mode in MDVS was overlooked in the three MDVS models [9,11,3].

5 An Efficient and Constant-Size MDVS

Another shortcoming in Chow's scheme [3] (including our revised scheme) is that signature length is proportional to number of verifiers. For a large set of verifiers, the length of a multi-designated verifier signature (growing linearly with the size) will be impractical. In this section, we propose a new constant-size MDVS scheme. Our scheme is based on Boneh-Boyen short signature scheme [1]. Furthermore, our scheme does not rely on the random oracle model for its proof of security. The description of our new construction is as follows.

- Setup: Let $(\mathbb{G}_1, \mathbb{G}_2)$ be a bilinear groups where $|\mathbb{G}_1| = |\mathbb{G}_2| = p$, k be the system security parameter and g be the generator of $\mathbb{G}_1$. e denotes the bilinear pairing $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. The system parameter $cp = \{\mathbb{G}_1, \mathbb{G}_2, p, k, e, g\}$ which is shared by all the users in the system.
- KeyGen: Given cp, Signer picks two secret numbers $u_a, v_a \in_R Z_p^*$ and set secret key $sk_s = (u_a, v_a)$. Signer's public key is $pk_s = (U_a, V_a) = (g^{u_a}, g^{v_a})$. Similarly, for $i = 1, \cdots, n$, the verifier V_i's secret/public key pair is $sk_{v_i} = u_i, pk_{v_i} = g^{u_i}$.
- Sign: Given a message m, signer's secret key sk_s and verifiers' public keys pk_{v_i}, signer chooses $r, s \in Z_p^*$ and computes $Q_1 = g^{\frac{s}{u_a+m+v_a r}}$, $Q_2 = (\prod_{i=1}^n pk_{v_i})^s$, $Q_3 = g^s$. The designated verifier signature of m is $\sigma = (r, Q_1, Q_2, Q_3)$.
- Verify: Given message-signature pair (m, σ), the signer's public key pk_s, verifier's public key pk_{v_i}, the verifier checks whether
  1. $e(Q_1, U_a \cdot g^m \cdot V_a^r) = e(Q_3, g)$;
  2. $e(\prod_{i=1}^n pk_{v_i}, Q_3) = e(Q_2, g)$.

  If all the equalities hold, output Accept, otherwise Reject.

The correctness of the scheme is straightforward. The non-transferability property is established by the following theorem.

Theorem 1. *The new scheme satisfies non-transferability.*

Similar to theorem in [18,17], the next theorem shows the unforgeability of the scheme under the strong Diffie-Hellman assumption (SDH)[1] and the knowledge-of-exponent (KEA) assumption [2].

Theorem 2. *The MDVS scheme achieves existential unforgeable against adaptive chosen message attack (EUF-CMA) security under strong Diffie-Hellman assumption and knowledge-of-exponent assumption.*

Due to page limitation, the details of formal security proof and some remarks are provided in the full version.

6 Conclusions

In this paper, we presented a generic definition of designated verifier signature (DVS) and examined several security notions in DVS and its three main variants, namely strong DVS, multi-DVS and universal DVS. Two concrete DVS schemes (UDVS and MDVS) was respectively analyzed from delegatability and forgeability. We proposed the refined classification of MDVS models and indicated that the three MDVS models [9,11,3] are unique. Finally, we also presented a new and efficient constant-size MDVS scheme, which is proven secure without incorporating the random oracle model.

Acknowledgements. We thank Helger Lipmaa and Raylin Tso for helpful and inspired discussions on DVS, as well as the anonymous referees for useful comments.

References

1. Boneh, D., Boyen, X.: Short Signatures Without Random Oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004)
2. Bellare, M., Palacio, A.: The Knowledge-of-Exponent Assumptions and 3-Round Zero-Knowledge Protocols. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 273–289. Springer, Heidelberg (2004)
3. Chow, S.S.M.: Identity-based Strong Multi-Designated Verifiers Signatures. In: Atzeni, A.S., Lioy, A. (eds.) EuroPKI 2006. LNCS, vol. 4043, pp. 257–259. Springer, Heidelberg (2006)
4. Goldwasser, S., Micali, S., Rivest, R.L.: A Digital Signature Scheme Secure Against Adaptive Chosen Message Attacks. SIAM Journal on Computing 17(2), 281–308 (1988)
5. Huang, X., Susilo, W., Mu, Y., Zhang, F.: Restricted Universal Designated Verifier Signature. In: Ma, J., Jin, H., Yang, L.T., Tsai, J.J.-P. (eds.) UIC 2006. LNCS, vol. 4159, pp. 874–882. Springer, Heidelberg (2006)
6. Jakobsson, M., Sako, K., Impagliazzo, R.: Designated Verifier Proofs and Their Applications. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 143–154. Springer, Heidelberg (1996)
7. Li, Y., Lipmaa, H., Pei, D.: On Delegatability of Four Designated Verifier Signatures. In: Qing, S., Mao, W., Lopez, J., Wang, G. (eds.) ICICS 2005. LNCS, vol. 3783, pp. 61–71. Springer, Heidelberg (2005)

8. Laguillaumie, F., Vergnaud, D.: Designated Verifier Signatures: Anonymity and Efficient Construction from Any Bilinear Map. In: Blundo, C., Cimato, S. (eds.) SCN 2004. LNCS, vol. 3352, pp. 105–119. Springer, Heidelberg (2005)
9. Laguillaumie, F., Vergnaud, D.: Multi-designated Verifiers Signatures. In: Lopez, J., Qing, S., Okamoto, E. (eds.) ICICS 2004. LNCS, vol. 3269, pp. 495–507. Springer, Heidelberg (2004)
10. Lipmaa, H., Wang, G., Bao, F.: Designated Verifier Signature Schemes: Attacks, New Security Notions and A New Construction. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 459–471. Springer, Heidelberg (2005)
11. Ng, C.Y., Susilo, W., Mu, Y.: Universal Designated Multi Verifier Signature Schemes. In: SNDS 2000, pp. 305–309. IEEE Press, NJ, New York (2005)
12. Steinfeld, R., Bull, L., Wang, H., Pieprzyk, J.: Universal Designated-Verifier Signatures. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 523–542. Springer, Heidelberg (2003)
13. Saeednia, S., Kremer, S., Markowitch, O.: An Efficient Strong Designated Verifier Signature Scheme. In: Lim, J.-I., Lee, D.-H. (eds.) ICISC 2003. LNCS, vol. 2971, pp. 40–54. Springer, Heidelberg (2004)
14. Susilo, W., Wu, W., Mu, Y., Huang, X.: On the "Non-Delegatability" Notion of Designated Verifier Signature Schemes. In: IWAP 2006, Springer-Verlag, Heidelberg (to appear, 2006)
15. Susilo, W., Zhang, F., Mu, Y.: Identity-Based Strong Designated Verifier Signature Schemes. In: Wang, H., Pieprzyk, J., Varadharajan, V. (eds.) ACISP 2004. LNCS, vol. 3108, pp. 313–324. Springer, Heidelberg (2004)
16. Tso, R., Okamoto, T., Okamoto, E.: Practical Strong Designated Verifier Signature Schemes Based on Double Discrete Logarithms. In: Feng, D., Lin, D., Yung, M. (eds.) CISC 2005. LNCS, vol. 3822, pp. 113–127. Springer, Heidelberg (2005)
17. Vergnaud, D.: New Extensions of Pairing-based Signatures into Universal Designated Verifier Signatures. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 58–69. Springer, Heidelberg (2006)
18. Zhang, R., Furukawa, J., Imai, H.: Short Signature and Universal Designated Verifier Signature Without Random Oracles. In: Ioannidis, J., Keromytis, A.D., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 483–498. Springer, Heidelberg (2005)
19. Zhang, F., Susilo, W., Mu, Y., Chen, X.: Identity-based Universal Designated Verifier Signatures. In: Enokido, T., Yan, L., Xiao, B., Kim, D., Dai, Y., Yang, L.T. (eds.) Embedded and Ubiquitous Computing – EUC 2005 Workshops. LNCS, vol. 3823, pp. 825–834. Springer, Heidelberg (2005)