Delegatability of Designated Verifier Signature

Yongjian Liao
School of Computer Science & Engineering
University of Electronic Science & Technology of China
Chengdu, China
liaoyj@uestc.edu.cn

Zhiguang Qin
School of Computer Science & Engineering
University of Electronic Science & Technology of China
Chengdu, China
Qinzg@uestc.edu.cn

***Abstract*—There are only two possible signers in designated verifier signature(DVS) scheme, thus anyone else can not know who is the real signer according to signature/message pairs. Lipmaa et al. discovered delegatability attack on almost all existing designated verifier signature according to original definition of DVS, and later Li et al. subdivided the delegation and defined verifier-only delegatability. Here, we point out the formal definition of verifier-only delegatability is not reasonable and redefine it. Meanwhile we show ZFI DVS scheme is not verifier-only delegatable, but is delegatable, and show ZJ DVS scheme is verifier-only delegatable scheme. We present notion of signer-only delegatability and put forward general construction from verifier-only delegatable DVS scheme to signer-only delegatable DVS scheme, vice verse. We using ZJ DVS scheme as example to show how to construct signer-only delegatable scheme. Finally we classify delegatable DVS scheme into both signer and designated verifier delegatable scheme, verifier-only delegatable scheme and signer-only scheme.**

***Index Terms*—signature; designated verifier signature; delegatability.**

I. Introduction

Designated verifier signature (DVS) scheme [2] enables a signer to sign a message so that only the designated verifier can be convinced with the authenticity of the signature (although anyone can verify the signature publicly). Hence, the designated verifier cannot transfer the conviction to others because he himself is able to generate signatures according to a distribution that is computationally or statistically close to the distribution of signatures, generated by the signer. Additionally, nobody else but the signer and the designated verifier can generate valid signatures. This concept was introduced to complement the notion of undeniable signatures.

Since 2003 a lot of DVS schemes were presented. However, Lipmaa, Wang and Bao[6] discovered a new attack for DVS in 2005 according to original definition of DVS, called the delegatability (signing rights), and revisited the DVS security. They identified a new security property for DVS non-delegatability, and showed that several previously proposed DVS schemes [4], [9], [8], [10] were delegatable. Later, Li, Lipmaa and Pei[3] showed that signing rights of DVS in papers [11], [7], [12], [5] were delegatable. Therefore, only the DVS schemes [2], [6] remain to be considered as "secure". In the same paper, they defined the notion of verifier-only delegatability, which is a little difference with the original one[6]. They showed ZFI scheme[12] and LV multi-DVS scheme[5] were verifier-only delegatable.

Lipmaa et al.'s original ideas are that the side information can be disclose to anyone(not exclude the signer and designated verifier). However, some of Li et al.'s ideas are that this information must not disclose to them(signer and designated verifer). For example, Li et al. require two designated verifiers collude to leak sum of their secret keys to a third party to produce delegation. Therefor if any designated verifier of the two knows the sum, then this implies he can compute the other one's secret key. Thus delegatability of multi-designated verifiers signature scheme is more complex than DVS schemes, and seems not to be consistent with Lipmaa et al.'s delegatability. To simplify notion in entire paper we only investigate delegatability of DVS schemes.

Our contributions: Verifier-only delegatability is special delegatability defined by Lipmaa et la.[6]. But verifier-only delegatability defined by Li et al. requires that it is not delegatable first for both signer and designated verifier. This is inconsistent with the original idea. Furthermore, definition of verifier-only delegatability seems to apply mechanically the definition of non-delegatability in paper [6], this is another error. Thus we redefine concept of verifier-only delegatability and show ZFI scheme is delegatable but not verifier-only delegatable. we also show ZJ scheme is verifier-only delegatable.

Meanwhile, we put forward notion of signer-only delegatability, and show how to transform verifier-only delegatable scheme to signer-only delegatable scheme, vice verse. Thus we classify delegatable schemes into three possible types: both signer and designated verifier delegatable schemes, verifier-only delegatable schemes and signer-only delegatable schemes.

II. Delegatability

A. Delegatability of Signing Rights

Motivated to the original informal definition of DVS[2], Lipmaa, Wang and Bao presented a non-standard attack, which is called delegating attack. In fact, they pointed out some flaws in almost all (strong) DVS schemes. The details are: Signer can delegate his signing ability — with respect to a fixed designated verifier — to a third party T, without revealing his secret key or making it possible for T to sign with respect to other designated verifiers(verifiers can also do it). This property is considered as a serious weakness since this conflicted with the original informal definition of DVS in [2]. In the delegatable DVS scheme signer or designated verifier is able to generate some side information which any one obtains

978–1–4244–1674–5/08/$25.00 © 2008 IEEE

can generate valid signature, and which is independent on the message/signature pairs.

Definition 1. Non-delegatability of signing rights [6]: Let $\kappa \in [0, 1]$ be the knowledge error. Δ is (τ, κ)-non-delegatable if there exists a black-box knowledge extractor K that, for every algorithm $\mathcal{F}$ and for every valid signature σ, satisfies the following condition: For every (sk_S, pk_S), (sk_D, pk_D) generated by $KeyGen$, and message m, if $\mathcal{F}$ produces a valid signature on m with probability $\varepsilon > \kappa$, then on input m and on access to the oracle $\mathcal{F}_m$, K produces one of the secret keys (sk_S, sk_D) in expected time $\frac{\tau}{\varepsilon - \kappa}$(without counting the time to make the oracle queries). Here, $\mathcal{F}$'s probability is taken over the choice of her random coins and over the choice of hash function H, $\mathcal{F}_m$denotes $\mathcal{F}$ with m as its input.

Non-delegatability of signing rights is very strong property, which requires any side information(except secret keys of both parties) in DVS schemes is not able to generate valid signatures. The weak concepts is non-delegatbility of signing rights for designated verifier(or signer). Li et al.[3] originally paid attention to verifier-only delegability attack, However there is not DVS scheme satisfying this property[1] (we will show ZFI is delegatable, but not verifier-only delegatable next). In theory, there exist four probabilities for DVS schemes: delegatability, verifier-only delegatability, signer-only delegatability and non-delegatability. All DVS schemes attacking in paper [6] belong to the first type, and the last one is secure DVS scheme like JSI scheme[2] and LWB scheme[6]. The spare two are special cases of the first. We informally define them as follow.

Delegatability: Both signer and designated verifier can delegate the signing rights to a third party without disclosing their secret keys .

Verifier-only delegatability: Designated verifier can delegate the signing rights to a third party without disclosing his secret key, which the signer cannot do it.

Signer-only delegatability: Signer can delegate the signing rights to a third party without disclosing their secret key, which the designated verifier cannot do it.

B. Formal Definition

In this section, we formally redefine the verifier-only delegatability by defining non-verifier-only delegatability.

Definition 2. Non-verifier-only-delegatability of signing rights: Let $\kappa \in [0, 1]$ be the knowledge error. Δ is (τ, κ)-non-verifier-only delegatability if there exists a black-box knowledge extractor K that, for every algorithm $\mathcal{F}$ and for every valid signature σ, satisfies the following condition: For every (sk_{S_i}, pk_{S_i}), (sk_D, pk_D) $i \in \{1, \cdots, n\}$ generated by $KeyGen$, and message m, if $\mathcal{F}$ produces a valid signature on m with probability $\varepsilon > \kappa$, then on input m and on access to the oracle $\mathcal{F}_m$, K produces the designated verifier's secret keys sk_D in expected time $\frac{\tau}{\varepsilon - \kappa}$(without counting the time to make the oracle queries). Here, $\mathcal{F}$'s probability is taken over the choice of her random coins and over the choice of hash function H, $\mathcal{F}_m$denotes $\mathcal{F}$ with m as its input.

This definition means that non-verifier-only delegatable scheme satisfies following conditions: if there exists some side information which is used to generate valid signature for any signers, then it will output verifier's secret key. In contrast, there exists some side information to generate valid signature which is only product by verifier in verifier-only delegatable scheme.

Remark 3. This definition is slightly different to the LWB's one. Because delegatable schemes exist delegatable side information of fixed signer w.r.t. fixed designated verifier(of course, there also exist other possibilities), while every verifier-only delegatable scheme exists delegatble side information of any signer w.r.t. fixed designated verifiers.

III. Verifier-only Delegatability

Li et al. found out ZFI DVS scheme was verifier-one delegatable scheme. But we will show that this scheme is delegatable for both signer and designated verifier. We describe a slightly simplified version of ZFI DVS scheme as follows firstly.

- $Setup$: Choose a bilinear group pair (G_1, G_2) of prime order $G_1 = G_2 = q$, with a bilinear map e: $G_1 \times G_2 \rightarrow G_T$ and an isomorphism ψ: $G_2 \rightarrow G_1$. Here G_1 is multiplicative group. Choose a random generator $g_2 \in G_2$, and compute $g_1 \leftarrow \psi(g_2) \in G_1$. Then the common parameter is $param \leftarrow (q, G_1, G_2, G_T, e, \psi, g_1, g_2)$.
- $KeyGen(param)$: Pick random x, $y \in Z_q$, compute $u \leftarrow g_2^x$, $v \leftarrow g_2^y$. The public key is $PK \leftarrow (u, v)$ and the secret key is $SK \leftarrow (x, y)$. In particular, S has a key pair with $PK_S \leftarrow (u_S, v_S)$, $SK_S \leftarrow (x_S,\ y_S)$ and D has a key pair with $PK_D \leftarrow (u_D, v_D)$, $SK_D \leftarrow (x_D,\ y_D)$.
- $Sign_{SK_S,PK_D}(m)$: Pick a random $r \longrightarrow Z_q$. If $x_S + r + y_S m \neq 0$ mod q,, compute $\sigma' \leftarrow g_1^{1/(x_S+r+y_S m)}$, $h \leftarrow g_2^r$, $d \leftarrow e(\psi(u_D), v_D^r)$. Return $\sigma \leftarrow (\sigma', h, d)$.
- $Simul_{PK_S,SK_D}(m)$: Pick a random $s \in Z_q$ and compute $\sigma' \leftarrow g_1^s$, $h \leftarrow g_2^{1/s} u_S^{-1} v_S^{-m}$ and $d \leftarrow e(g_1, h)^{x_D y_D}$. $\sigma \leftarrow (\sigma', h, d)$.
- $Verify_{PK_S,SK_D}(\sigma, h, d)$: Output accept if $e(g_1,\ g_2) = e(\sigma',\ u_S h v_S^m)$ and $d = e(\psi(u_D), h^{y_D})$. Otherwise, output reject.

In paper [3], Li et al. claimed that this scheme was verifier-only delegatable according to reveal $g_1^{x_D y_D}$ and it does not depend on the signer. However, we show that signer is also able to delegate signing rights to a third party. Signer can disclose $\alpha \leftarrow g_2^{x_S y_D}$ and $\beta \leftarrow g_2^{y_S y_D}$(or $g_2^{x_S x_D}$ and $g_2^{y_S x_D}$). Anyone who obtain $(\alpha,\ \beta)$ is able to create valid signature as follows.

Randomly choose $s \in Z_q$ and compute $\sigma' \leftarrow g_1^s$, $h \leftarrow g_2^{1/s} u_S^{-1} v_S^{-m}$ and $d \leftarrow e(\psi(g_2^{x_D}),\ v_D^{1/s}\ \alpha^{-1}\ \beta^{-m})$. $\sigma \leftarrow (\sigma', h, d)$.

$$\begin{aligned} e(\sigma', u_S h v_S^m) &= e(g_1^s, u_S g_2^{1/s} u_S^{-1} v_S^{-m} v_S^m) \\ &= e(g_1^s, g_2^{1/s}) \\ &= e(g_1, g_2). \end{aligned}$$

[1] we only consider DVS scheme, but not multi-DVS scheme

$$\begin{aligned} e(\psi(u_D), h^{y_D}) &= e(\psi(g_2^{x_D}), (g_2^{1/s} u_S^{-1} v_S^{-m})^{y_D}) \\ &= e(\psi(g_2^{x_D}), g_2^{y_D/s} u_S^{-y_D} v_S^{-y_D m}) \\ &= e(\psi(g_2^{x_D}), v_D^{1/s} \alpha^{-1} \beta^{-m}) \\ &= d. \end{aligned}$$

Thus, ZFI is one that both signer and designated verifier are able to disclose side information. However there exist differences between them, since delegation of signer only reveal side information which can produce valid signatures with respect to fixed signer with fixed designated verifier, in contrast, the side information which designated verifiers disclose can create valid signature with respect to any signers with fixed designated verifier. This means delegable capability between them is different.

IV. Signer-only Delegatability

We formally define the signer-only delegatability by defining non-signer-only delegatability.

Definition 4. Non-signer-only-delegatability of signing rights: Let $\kappa \in [0, 1]$ be the knowledge error. Δ is (τ, κ)-non-signer-only delegatability if there exists a black-box knowledge extractor K that, for every algorithm $\mathcal{F}$ and for every valid signature σ, satisfies the following condition: For every (sk_S, pk_S), (sk_{D_i}, pk_{D_i}) $i \in \{1, \cdots, n\}$ generated by $KeyGen$, and message m, if $\mathcal{F}$ produces a valid signature on m with probability $\varepsilon > \kappa$, then on input m and on access to the oracle $\mathcal{F}_m$, K produces the singer's secret keys sk_S in expected time $\frac{\tau}{\varepsilon - \kappa}$(without counting the time to make the oracle queries). Here, $\mathcal{F}$'s probability is taken over the choice of her random coins and over the choice of hash function H, $\mathcal{F}_m$denotes $\mathcal{F}$ with m as its input.

For any concrete DVS scheme, the different side information do different works. Maybe some side information is not any useful for signing a message. For example, anyone know $g^{x_S^2 x_D^2}$ can not produce valid signature of HSMZ scheme [1], but obtaining $g^{x_S x_D}$ is enough to generate valid signatures(the detail signature scheme is in paper [1]). Thus we think it is more reasonable to consider who produce delegation information than what exactly can be delegated.

Until now, we do not know which scheme is signer-only delegatable scheme. If these type schemes exist, then we classify delegatable DVS scheme as follow: both signer and designated verifier delegatable scheme, verifier-only delegatable scheme and signer-only delegatable scheme.

V. General Construction from Verifier-only Delegatability Scheme to Signer-only Delegatability one

In the section, we solve the problem leaving in previous section — whether does there exist signer-only delegatable schemes or not? We design general construction how to transform verifier-only delegatable scheme to signer-only delegatable one under assumption that the verifier-only delegatable DVS schemes exist. Actually the converse also holds.

Let DVS is verifier-only delegatable DVS scheme, and it consists of following four algorithms: $Setup$, $Sign_{sk_S,pk_D}$, $Simul_{sk_D,pk_S}$, and $Verify_{pk_D,pk_S}$. We construct $\overline{DVS}$ DVS scheme by transformation of keys's index.

- $\overline{Setup}$: The same as $Setup$.
- $\overline{Sign_{sk_S,pk_D}}$: To substitute sk_D for sk_S, pk_S for pk_D, and pk_D for pk_S in algorithm $Sign_{sk_S,pk_D}$ of DVS scheme.
- $\overline{Simul_{sk_D,pk_S}}$: To substitute sk_S for sk_D, pk_S for pk_D, and pk_D for pk_S in algorithm $Simul_{sk_D,pk_S}$ of DVS scheme.
- $\overline{Verify_{pk_D,pk_S}}$: To substitute pk_S for pk_D, and pk_D for pk_S in algorithm $Verify_{pk_D,pk_S}$ of DVS scheme.

$\overline{DVS}$ scheme is also a DVS scheme, it is easy to verify property of unforgeability and non-transferability if DVS scheme satisfies these two properties. This is only a little technique problem: the signer does not know designated verifier's secret key and designated verifier does not know the signer's secret key too. Since our transformation is only transformation of keys's index,signer in $\overline{DVS}$ scheme can perform signing algorithm by using $simul$ of DVS scheme, and this is analogous to $\overline{Simul}$. Thus if DVS is verifier-only delegatable scheme, then $\overline{DVS}$ is signer-only delegatable scheme.

VI. Concrete Example

Recently, Zhang and Ji [13] presented a new DVS scheme without random oracle from pairings. However, it is a verifier-only delegatable scheme. We first review their scheme as follow.

- $Setup$: Let (G_1, G_2) be two cyclic groups of prime order p. g_1 is a generators of G_1 and g_2 is a generator of G_2. An isomorphism ψ: $G_2 \rightarrow G_1$, with $\psi(g_2) \rightarrow g_1$. We also assume that the message m to be signed is an element in Z_p . The signer, Alice, randomly chooses $x_S \in Z_p$ and computes the corresponding public key $y_S \leftarrow g_2^{x_S}$, then selects a generator u_2 as the partial public key. The designated verifier, Bob, also selects his secret key $x_D \in Z_q$ and sets his public key $y_D \leftarrow g_2^{x_D}$. Finally, publish Alice's public key (y_S, u_2) and Bob's public key y_D.
- $Sign$: To sign a message m for Bob, Alice performs the following steps: . randomly choose two numbers r, $s \in Z_p$. . first compute $\tau \leftarrow g_2^s$; then compute $\sigma' \leftarrow (g_1^m u_1 y_D^s)^{1/(x_S+r)}$ where $u_1 \leftarrow \psi(u_2)$. In the unlikely event that $x_S + r = 0 \bmod p$, we try again with a different random r. Finally, the resultant signature on message m is $\sigma \leftarrow (\sigma', r, \tau)$
- $Simul$: Bob can produce a signature σ on arbitrary message m intended for himself, by performing the follow steps: randomly choose two numbers r'', α' $\in Z_p$ and the signed message m. compute $\tau' \leftarrow y_S^{\alpha' x_D^{-1}} g_2^{-m x_D^{-1}} u_2^{-x_D^{-1}} g_2^{r'' \alpha'}$ and $\sigma' \leftarrow g_1^{\alpha'}$. The signature σ on the message m is $(\sigma', \tau, r' = r'' x_D)$.
- $Verify$: Given a signature $\sigma \rightarrow (\sigma', \tau, r)$ on the message m, Bob verifies as follows:

$$e(\sigma', y_S g_2^r) = e(g_1, g_2^m u_2 \tau^{x_D})$$

If the above equation holds, then it denotes that the signature is valid. Otherwise, output reject.

This DVS scheme is real verifier-only delegatable. The designated verifier discloses $(y_S^{x_D^{-1}}, g_2^{x_D^{-1}}, u_2^{-x_D^{-1}})$ to a third party T, then T can create valid signature as follows.

Randomly choose r', $\alpha' \in Z_p$ and message m, compute $\tau' \leftarrow y_S^{\alpha' x_D^{-1}} g_2^{-m x_D^{-1}} u_2^{-x_D^{-1}} g_2^{x_D^{-1} r' \alpha'}$ and $\sigma' \leftarrow g_1^{\alpha'}$. The signature σ on the message m is (σ', τ', r').

$$\begin{aligned}
e(g_1, g_2^m u_2 \tau'^{x_D}) &= e(g_1, g_2^m u_2 (y_S^{\alpha' x_D^{-1}} g_2^{-m x_D^{-1}} u_2^{-x_D^{-1}} g_2^{x_D^{-1} r' \alpha'})^{x_D}) \\
&= e(g_1, g_2^m u_2 y_S^{\alpha'} g_2^{-m} u_2^{-1} g_2^{r'\alpha'}) \\
&= e(g_1, (y_S g_2^{r'})^{\alpha'}) \\
&= e(\sigma', y_S g_2^{r'}).
\end{aligned}$$

So the signature σ is valid signature of message m. It seems there doesn't exist signer's delegatability. Since given pk_D and any two of σ', τ', r', it will be up against solving some hard problems which are dependent on message(the delegatable information must be independent on message). The details are following.

Given τ, r, to compute σ' must know x_S so far; and given σ', r, to compute τ must know x_D; finally, r is part of signature, and which is random, so given τ, r to compute σ' must face to solve strong Diffie-Hellman problem(fixed r, it is able to generate delegation, but this case is not consistent with randomly picking r). Thus until now ZJ DVS scheme can be viewed as verifire-only delegatable DVS scheme.

A. The First Signer-only Delegatable scheme

- $Setup$: The same to ZJ scheme.
- $Sign$: To sign a message m, signer performs the following steps: Randomly choose two numbers r, $s \in Z_p$. first compute $\tau \leftarrow g_2^s$; then compute $\sigma' \leftarrow (g_1^m u_1 y_S^s)^{1/(x_D+r)}$ where $u_1 \leftarrow \psi(u_2)$. In the unlikely event that $x_S + r =$ 0 mod p, we try again with a different random r. Finally, the resultant signature on message m is $\sigma \leftarrow (\sigma', r, \tau)$
- $Simul$: Verifier can produce a signature σ on arbitrary message m, by performing the follow steps: Randomly choose two numbers r'', $\alpha' \in Z_p$ and the signed message m. compute $\tau' \leftarrow y_D^{\alpha' x_S^{-1}} g_2^{-m x_S^{-1}} u_2^{-x_S^{-1}} g_2^{r''\alpha'}$ and $\sigma' \leftarrow g_1^{\alpha'}$. The signature σ on the message m is $(\sigma', \tau, r' = r'' x_S)$.
- $Verify$: Given a signature $\sigma \rightarrow (\sigma', r, \tau)$ on the message m, Bob verifies as follows:

$$e(\sigma', y_D g_2^r) = e(g_1, g_2^m u_2 \tau^{x_S})$$

If the above equation holds, then it denotes that the signature is valid. Otherwise, output reject.

It is easy to verify this scheme. There is only a little performing problem. In fact, signer performs signing algorithm by using ZJ's simulation algorithm and the designated verifier performs simulation algorithm by using ZJ's signing algorithm. It is easy to verify The signer discloses $(y_D^{x_S^{-1}}, g_2^{x_S^{-1}}, u_2^{-x_S^{-1}})$ to a third party T, then T can create valid signature as follows. So this scheme is signer-only delegatable scheme.

VII. Conclusion

In this paper, we investigate types of probability of delegatability for DVS schemes(not include multi-designated verifier) in theory and correct some existing error on delegatability. We first point out the definition of verifier-only delegatability exists errors and redefine it. Then we show ZFI scheme is delegatable but not verifier-only delegatable. we also show ZJ scheme is verifier-only delegatable. Meanwhile, we present notion of signer-only delegatability, and show how to transform verifier-only delegatable scheme to signer-only delegatable scheme, vice verse. We use ZJ DVS scheme as example to show how to construct signer-only delegatable scheme. Finally we classify delegatable schemes into both signer and designated verifier delegatable scheme, verifier-only delegatable scheme and signer-only delegatable scheme.

Acknowledgment

The authors would like to thank the anonymous referees for useful references. And this work is supported by the National Natural Science Foundation of China under contract no. 60673075 and the National High Technology Research and Development Program of China (863) under contract no. 2006AA01Z428.

References

[1] X. Huang, W. Susilo, Y. Mu and F. Zhang, *Short (Identity-based)Strong Designated Verifier Signature Schemes*. ISPEC'06, pp.214-225, 2006.

[2] M. Jakobsson, K. Sako and R. Impagliazzo *Designated verifier proofs and their applications*. Advances in Cryptology-EuroCrypt'96, LNCS 1070, pp. 143-154, 1996.

[3] Y. Li, H. Lipmaa and D. Y. Pei,*On Delegatability of Four Designated Verifier Signatures*. ICICS 2005, pp.61-71, 2005.

[4] F. Laguillaumie and D. Vergnaud, *Designated Verifier Signatures: Anonymity and Efficient Construction from Any Bilinear Map*. SCN 2004, pp.105-119, 2004.

[5] F. Laguillaumie and D. Vergnaud, *Multi-designated Verifiers Signatures*. ICICS 2004, pp.495-507, 2004.

[6] H. Lipmaa, G. Wang and F. Bao, *Designated Verifier Signature Schemes: Attacks, New Security Notations and a New construction*. ICALP 2005, pp.459-471, 2005.

[7] C. Y. Ng, W. Susilo and Y. Mu, *Universal Designated Multi Verifier Signature Schemes*. The International Workshop on Security in Networks and Distributed Systems (SNDS 2005), pp.305-309, 2005.

[8] R. Steinfeld, L. Bull, H. Wang and J. Pieprzyk, *Universal Designated-Verifier Signatures*. LNCS 2894, pp. 523-542, 2003.

[9] S. Saeednia, S. Kremer and O. Markowitch, *An efficient strong designated verifier signature scheme*. ICISC'03, pp.40-54, 2003.

[10] R. Steinfeld, H. Wang and J. Pieprzyk, *Efficient Extension of Standard Schnorr/RSA Signatures into Universal Designated-Verifier Signatures*. PKC'04, pp.86-100, 2004. ,

[11] W. Susilo, F. Zhang and Y. Mu, *Identity-based Strong Designated Verifier Signature Schemes*. ACISP'04, pp. 313-324, 2004.

[12] R. Zhang, J. Furukawa and H. Imai,*Short Signature and Universal Designated Verifier Signature Without Random Oracles*. ACNS 2005, LNCS 3531, pp.483-498, 2005.

[13] J. Zhang and C. Ji, *An Efficient Designated Verifier Signature Scheme without Random Oracles*. First International Symposium on Data, Privacy and E-Commerce, pp.338-340, 2007.