JAIST Repository
https://dspace.jaist.ac.jp/

Title	An Anonymous Designated Verifier Signature Scheme with Revocation: How to Protect a Company's Reputation
Author(s)	Emura, Keita; Miyaji, Atsuko ; Omote, Kazumasa
Citation	Lecture Notes in Computer Science, 6402: 184-198
Issue Date	2010
Type	Journal Article
Text version	author
URL	http://hdl.handle.net/10119/9511
Rights	This is the author-created version of Springer, Keita Emura, Atsuko Miyaji and Kazumasa Omote, Lecture Notes in Computer Science, 6402, 2010, 184-198. The original publication is available at www.springerlink.com, http://dx.doi.org/10.1007/978-3-642-16280-0_12
Description	Provable Security, 4th International Conference, ProvSec 2010, Malacca, Malaysia, October 13-15, 2010. Proceedings

Japan Advanced Institute of Science and Technology

An Anonymous Designated Verifier Signature Scheme with Revocation: How to Protect a Company's Reputation

Keita Emura[1], Atsuko Miyaji[2], and Kazumasa Omote[2]

[1] Center for Highly Dependable Embedded Systems Technology
[2] School of Information Science
Japan Advanced Institute of Science and Technology, 1-1, Asahidai, Nomi, Ishikawa, 923-1292, Japan
{k-emura, miyaji, omote}@jaist.ac.jp

Abstract. There are many cryptographic schemes with anonymity, such as group signatures. As one important property, anonymity revocation has been introduced. In such schemes, the fact of *whether a signer's rights have been revoked or not* is important additional information. For example, if a third party knows that there are many revoked members in a company, then the company's reputation may be damaged in many ways. People may think that *there might be many problematic employees (who have bad behavior-s) in this company*, *there might be many people who have quit, i.e., the labor environment may not be good*, and so on. To avoid such harmful rumors, in this paper, we propose an Anonymous Designated Verifier Signature (ADVS) scheme with revocation. In ADVS, a designated verifier can only verify a signature anonymously, and a third party cannot identify whether the rights of the signer have been revoked or not. We show two security-enhanced schemes as applications of our scheme: a biometric-based remote authentication scheme, and an identity management scheme.

1 Introduction

Back Ground: There are many cryptographic schemes with anonymity, such as group signatures [6]. Anonymous schemes are useful to protect a signer's privacy, and therefore many applications of group signature have been proposed such as the BCPZ (Bringer, Chabanne, Pointcheval, and Zimmer) biometric-based authentication scheme [5], the IMSTY (Isshiki, Mori, Sako, Teranishi, and Yonezawa) identity management scheme [11], and so on. As one important property, anonymity revocation has been introduced [3, 4, 15, 17, 18]. In these revocable group signature schemes, revocation check can be executed by *any entity*. Actually, the fact of *whether a signer's rights have been revoked or not* is important additional information. Let a signatory group of a group signature scheme be a company. If a third party knows that there are many revoked members in this company, then the company's reputation may be damaged in many ways. For example, someone may think that:

- There might be many problematic employees (who have bad behavior) in this company.
- There might be many people who have quit, i.e., the labor environment may not be good.

In addition, there are possibilities of user privacy exposure, for example:

- If the third party knows an employee who left the company three days ago, and also knows a signer was revoked three days ago, then the signer may be this employee.

Actually, the third party can detect whether a signer was revoked or not by checking whether a value was added to a revocation list RL or not. In this example, the third party can link signatures made by this employee who has left by executing the revocation check, even if a group signature scheme with backward unlinkability (such as [15, 18]) is used[3]. This scenario can occur, since (revocable) group signatures are applied in many applications. As a solution for protecting against damage caused by rumors, we consider to apply a cryptographic primitive with a property that a third party cannot check whether a signer's rights have already been revoked or not. Someone may think that group signature schemes with Verifier-Local Revocation (VLR) [4, 15, 18] can be applied for this purpose. By hiding a revocation list RL from the third party[4], the third party can be prevented from executing the revocation check. However, there is a problem in this scenario: a revoked user can make a *valid group signature* which is verified by the third party, since the third party can verify the validity of this signature by using a group public key gpk only (RL is used for the revocation check only). Therefore, VLR group signature schemes are not useful in protecting the company's reputation. This suggests that it is not enough to restrict the revocation check. As another solution for protecting against damage caused by rumors, we need to apply a cryptographic primitive with properties that not only the third party cannot check whether a signer's rights have already been revoked or not, but also the third party cannot check whether a signature is valid or not. As a candidate for this purpose, Designated Verifier Signature (DVS) [7, 10, 13, 14, 16, 20–22] is nominated, since a signer can indicate a designated verifier. Especially, strong DVS has been proposed [13, 14] which enables protection of the signer's anonymity from a third party. However, in the verification phase of strong DVS, a designated verifier verifies a signature with *the public key of a signer* and the secret key of the designated verifier. This means that these schemes do not provide the signer anonymity from the designated verifier, and this is a difference between DVS and group signatures. In addition, DVS does not have the revocation property. To sum up, no previous group signature and DVS schemes can be applied to protect the company's reputation.

[3] Note that backward unlinkability means that even after a signer's rights are revoked, signatures made by the signer before the revocation remain anonymous.

[4] In VLR schemes, a verifier verifies a group signature by using a group public key gpk, and checks whether the rights of the signer have been revoked or not by using RL. A signer does not have to obtain RL to sign.

Our Contribution: In this paper, by applying the designated verification property of DVS, we propose a way to protect the company's reputation. By indicating a designated verifier, (1) a third party cannot check whether a signature is valid or not, and (2) the third party cannot check whether a signer's rights have already been revoked or not, and (3) no entity (except the opening manager OM, which is defined later) can determine who a signer is. We call this signature primitive Anonymous Designated Verifier Signature (ADVS) scheme with revocation. We compare these functions with other primitives in Table 1.

Table 1. Function Comparisons

	Signer Anonymity	Designated Verification	Designated Revocation Check
DVS [12]	no	yes	no
Strong DVS [13, 14]	yes*	yes	no
Revocable Group Signature [3, 4, 15, 17, 18]	yes	no	no
Our ADVS	yes	yes	yes

∗ From a third party only

The property (1) is the same concept as in DVS schemes. The property (2) is a difference between revocable group signatures and our scheme. As a difference between strong DVS and our signer-anonymous DVS scheme, our scheme protects the signer anonymity from the designated verifier (property (3)). We provide formal definitions of ADVS, and prove our scheme along with these definitions. Our ADVS scheme can be applied to *protecting company's reputation* scenario.

Related works: The concept of designated verifier proof was introduced in Jakobsson, Sako, and Impagliazzo [12] (called JSI scheme), where a specific designated verifier can only verify the validity of proofs made by a prover's secret key and a verifier's public key. In the JSI scheme, although any entity can verify the validity of a proof, this entity cannot distinguish whether the proof was made by a prover or not. The designated verifier can make the same proof, and only the prover and the designated verifier know who is the actual prover. The JSI scheme uses the *or proof technique* [8], namely, the actual signer knows the secret key of the signer *or* the secret key of the designated verifier. A DVS signature can be achieved [7] by using the ring signature scheme with a two-person group (namely, members are the signer and the designated verifier only). From the viewpoint of a third party, nobody knows who the actual signer is, although the third party can verify the signature. There are DVS schemes such that the validity of a signature can only be verified by a designated verifier by using his/her secret key (e.g., [10, 13, 22]). In these schemes, a third party cannot verify the validity of a signature. Designated revocation check property has been considered in [9]. However, that paper did not define formal security requirements, and there is a flaw whereby a designated verifier can link two signatures by using his/her secret key. A designated group signature scheme, which enables both signer anonymity and designated verifier property[5], has not been proposed yet.

Organization : The paper is organized as follows: Security definitions of ADVS are presented in Section 3. Our proposed ADVS scheme is described in Section 4. The security proofs are presented in Section 5. Applications of our ADVS scheme to the BCPZ biometric-based authentication scheme [5] and the IMSTY identity management scheme [11] are presented in Section 6.

2 Preliminary

In this section, we show definitions of bilinear groups and complexity assumptions. Note that $x \in_R S$ means x is randomly chosen for a set S.

2.1 Bilinear Groups

Definition 1. (**Bilinear Groups**) *Bilinear groups and a bilinear map are defined as follows:*

1. *$\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of prime order p.*
2. *g is a generator of $\mathbb{G}$.*
3. *e is an efficiently computable bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ with the following properties.*
   - *Bilinearity : for all $u, u', v, v' \in \mathbb{G}$, $e(uu', v) = e(u, v)e(u', v)$ and $e(u, vv') = e(u, v)e(u, v')$.*
   - *Non-degeneracy : $e(g, g) \neq 1_{\mathbb{G}_T}$ ($1_{\mathbb{G}_T}$ is the $\mathbb{G}_T$'s unit).*

2.2 Complexity Assumptions

Definition 2. (**DLIN assumption**) *[3] The Decision Linear (DLIN) problem in $\mathbb{G}$ is a problem, for input of a tuple $(u, v, h, u^\alpha, v^\beta, Z) \in \mathbb{G}^6$ where $\alpha, \beta \in \mathbb{Z}_p$ are random values, to decide whether $Z = h^{\alpha+\beta}$ or not. An algorithm $\mathcal{A}$ has advantage ϵ in solving DLIN problem in $\mathbb{G}$ if $Adv_{DLIN}(\mathcal{A}) := |\Pr[\mathcal{A}(u, v, h, u^\alpha, v^\beta, h^{\alpha+\beta}) = 0] - \Pr[\mathcal{A}(u, v, h, u^\alpha, v^\beta, h^z) = 0]| \geq \epsilon(\kappa)$, where $h^z \in \mathbb{G} \setminus \{h^{\alpha+\beta}\}$. We say that the DLIN assumption holds in $\mathbb{G}$ if no PPT algorithm has an advantage of at least ϵ in solving the DLIN problem in $\mathbb{G}$.*

Definition 3. (**q-SDH assumption**) *[2, 3] The q-Strong Diffie-Hellman (q-SDH) problem in $\mathbb{G}$ is a problem, for input of a $(q + 1)$ tuple $(g, g^\gamma, \cdots, g^{\gamma^q}) \in \mathbb{G}^{q+1}$ where $\gamma \in \mathbb{Z}_p$ is a random value, to compute a tuple $(x, g^{1/(\gamma+x)}) \in \mathbb{Z}_p \times \mathbb{G}$. An algorithm $\mathcal{A}$ has an advantage ϵ in solving the q-SDH problem in $\mathbb{G}$ if $\Pr[\mathcal{A}(g, g^\gamma, \cdots, g^{\gamma^q}) = (x, g^{1/(\gamma+x)})] \geq \epsilon$. We say that the q-SDH assumption holds in $\mathbb{G}$ if no PPT algorithm has an advantage of at least ϵ in solving the q-SDH problem in $\mathbb{G}$.*

[5] Note that the concept of designated group signature (called ML scheme) proposed in [16] is different from this concept: the ML scheme enables the verifier anonymity, where designated verifiers are indicated.

3 Definitions of ADVS

In this section, we define ADVS and its security requirements. The ADVS scheme consists of six algorithms, Setup, $\mathsf{KeyGen_S}$, $\mathsf{KeyGen_V}$, Sign, Verify, and Revoke. The group public key gpk and the group secret key gsk are obtained by executing $\mathsf{Setup}(1^\kappa)$, where κ is the security parameter. A signer public key spk and a signer secret key (which is also called a membership certificate) ssk are obtained by executing $\mathsf{KeyGen_S}(gpk, gsk)$. A verifier public key vpk and a verifier secret key vsk are obtained by executing $\mathsf{KeyGen_V}(1^\kappa)$. For a message M, a designated signature σ is obtained by executing $\mathsf{Sign}(gpk, ssk, vpk, M)$. σ is verified by executing $\mathsf{Verify}(gpk, vsk, M, \sigma)$. If both (1) σ is a valid signature, and (2) σ was made by using vpk (corresponding to vsk), then 1 is output, and 0, otherwise. A designated signature is *valid* means that (1) a signer has a membership certificate ssk issued by GM, and (2) the rights of the signer have not been revoked. Membership revocation is done by executing $\mathsf{Revoke}(gpk, gsk, ssk, RL)$, where RL is the revocation list. The Revoke algorithm outputs the updated RL. We assume three entities, the group manager GM, a signer, and a designated verifier, which runs (Setup, $\mathsf{KeyGen_S}$, Revoke), Sign, and ($\mathsf{KeyGen_V}$, Verify), respectively.

Next, we define the security requirements: *Unforgeability*, *Non-transferability*, and *Signer anonymity*. The DVS scheme is said to be unforgeable if the advantage is negligible for any probabilistic polynomial time (PPT) adversary $\mathcal{A}$ in the following experiment. In this experiment, $\mathcal{A}$ can access the signing oracle $\mathcal{O}_{\mathsf{Sign}(ssk^*, vpk)}$, where for an input message M, the signing oracle returns a signature σ made by ssk^* and designated to vpk, and appends (M, σ) to the set of signatures SigSet. In addition, $\mathcal{A}$ can access the verification oracle $\mathcal{O}_{\mathsf{Verify}(vsk)}$. For the input of the message/signature pair (M, σ), $\mathcal{O}_{\mathsf{Verify}(vsk)}$ returns the result of $\mathsf{Verify}(gpk, vsk, M, \sigma)$. In addition, $\mathcal{A}$ can access the corruption oracle $\mathcal{O}_{\mathsf{corr}}$. For the input of the identity of signer i, $\mathcal{O}_{\mathsf{corr}}$ returns ssk_i, and appends i to the set of corrupted users CU. Note that $\mathcal{A}$ cannot query i^* to the corruption oracle, where i^* is the target signer (who manages ssk^*). In addition, $\mathcal{A}$ can access the revocation oracle $\mathcal{O}_{\mathsf{revoke}}$. For the input of the identity of signer i, $\mathcal{O}_{\mathsf{revoke}}$ runs $\mathsf{Revoke}(gpk, gsk, ssk_i, RL)$. Note that $\mathcal{A}$ cannot query i^* to the revocation oracle. Finally, $\mathcal{A}$ outputs $(M^*, \sigma^*) \notin \mathsf{SigSet}$. To guarantee that no ssk_i ($i \in \mathsf{CU}$) were used to compute (M^*, σ^*), $\mathsf{Revoke}(gpk, gsk, ssk_i, RL)$ is executed for all corrupted users i.

Definition 4. *Unforgeability*

$$\begin{aligned}
Adv_{\mathcal{A}}^{UF}(\kappa) = \Pr\big[&(gpk, gsk) \leftarrow \mathsf{Setup}(1^\kappa);\ \mathsf{CU} \rightarrow \emptyset;\ \mathsf{SigSet} \rightarrow \emptyset;\ (vpk, vsk) \leftarrow \mathsf{KeyGen_V}(1^\kappa);\\
&(i^*, State) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Verify}(vsk)}(\cdot), \mathcal{O}_{\mathsf{corr}}(\cdot), \mathcal{O}_{\mathsf{revoke}}(\cdot)}(gpk, vpk);\\
&(spk^*, ssk^*) \leftarrow \mathsf{KeyGen_S}(gpk, gsk);\\
&(M^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Sign}(ssk^*, vpk)}(\cdot), \mathcal{O}_{\mathsf{Verify}(vsk)}(\cdot), \mathcal{O}_{\mathsf{corr}}(\cdot), \mathcal{O}_{\mathsf{revoke}}(\cdot)}(gpk, spk^*, vpk, State);\\
&\forall i \in \mathsf{CU},\ \mathsf{Revoke}(gpk, gsk, ssk_i, RL);\ (M^*, \sigma^*) \notin \mathsf{SigSet};\\
&\mathsf{Verify}(gpk, vsk, M^*, \sigma^*) = 1\big]
\end{aligned}$$

Next, we define Non-transferability. Non-transferability means that a designated verifier cannot produce evidence which convinces a third party that a signature was *actually* computed by the signer. The ADVS scheme is said to be non-transferable if the advantage is negligible for any PPT adversary $\mathcal{A}$ in the following experiment. Intuitively, there exists a simulated signing algorithm Sign' for which the distribution of $(M, \mathsf{Sign}(gpk, ssk, vpk, M))$ and the distribution of $(M, \mathsf{Sign}'(gpk, spk, vsk, M))$ are indistinguishable.

Definition 5. *Non-transferability*

$$\begin{aligned} Adv_{\mathcal{A}}^{\textit{Non-Trans}}(\kappa) = \big|\Pr\big[&(gpk, gsk) \leftarrow \mathsf{Setup}(1^\kappa);\ (spk, ssk) \leftarrow \mathsf{KeyGen_S}(gpk, gsk); \\ &(vpk, vsk) \leftarrow \mathsf{KeyGen_V}(1^\kappa); \\ &(M^*, State) \leftarrow \mathcal{A}(gpk, spk, ssk, vpk, vsk); \mu \in_R \{0,1\}; \\ &\sigma_0 \leftarrow \mathsf{Sign}(gpk, ssk, vpk, M^*);\ \sigma_1 \leftarrow \mathsf{Sign}'(gpk, spk, vsk, M^*); \\ &\mu' \leftarrow \mathcal{A}(\sigma_\mu, State); \mu = \mu'\big] - 1/2\big| \end{aligned}$$

Next, we define Signer anonymity. The ADVS scheme is said to be signer-anonymous if the advantage is negligible for any PPT adversary $\mathcal{A}$ in the following experiment. Intuitively, Signer anonymity means that $\mathcal{A}$ with vsk cannot determine who the actual signer is. This suggests that even if a malicious designated verifier opens its own secret key vsk, Signer anonymity is still effective.

Definition 6. *Signer anonymity*

$$\begin{aligned} Adv_{\mathcal{A}}^{\textit{Sign-Anon}}(\kappa) = \big|\Pr\big[&(gpk, gsk) \leftarrow \mathsf{Setup}(1^\kappa);\ (spk_0, ssk_0) \leftarrow \mathsf{KeyGen_S}(gpk, gsk); \\ &(spk_1, ssk_1) \leftarrow \mathsf{KeyGen_S}(gpk, gsk);\ (vpk, vsk) \leftarrow \mathsf{KeyGen_V}(1^\kappa); \\ &(M^*, State) \leftarrow \mathcal{A}(gpk, spk_0, ssk_0, spk_1, ssk_1, vpk, vsk) \\ &\mu \in_R \{0,1\}; \sigma_\mu \leftarrow \mathsf{Sign}(gpk, ssk_\mu, vpk, M^*); \\ &\mu' \leftarrow \mathcal{A}(\sigma_\mu, State); \mu = \mu'\big] - 1/2\big| \end{aligned}$$

4 The Proposed Scheme

In this section, we propose an Anonymous Designated Verifier Signature (ADVS) scheme with revocation. Let SPK be a Signature based on a Proof of Knowledge and $DSig(sigkey, M)$ be a digital signature of a message M under a signing key $sigkey$. $DSig(sigkey, M)$ is verified by using a verification key, $verkey$. We use $DSig(sigkey, M)$ to guarantee that GM updates RL. Intuitively, our construction is as follows: A signer computes an "or proof", namely, SPK with knowledge of either part-1: an actual signer knows the secret key of the signer (this is the short group signature proposed by Boneh et al. [3]), or part-2: the actual signer knows the secret key of a designated verifier. This construction is needed to achieve Non-transferability. In addition, the signer encrypts a part of the part-1 SPK using the public key of the designated verifier. We improve the revocation algorithm of the Nakanishi-Funabiki group signature [18] to satisfy the property that a third party cannot check whether a signer has already been revoked or not.

Protocol 1. *Our ADVS scheme*

$\mathsf{Setup}(1^\kappa)$**:** *Choose a prime number p, a bilinear group $(\mathbb{G}, \mathbb{G}_T)$ with order p, generators $g, h, u, v, f \in_R \mathbb{G}$, and $\gamma \in_R \mathbb{Z}_p$, and compute $\omega = g^\gamma$. Output $gpk = (e, (\mathbb{G}, \mathbb{G}_T), g, h, u, v, f, \omega, H, verkey)$ and $gsk = (\gamma, sigkey)$, where H is a cryptographic hash function from $\{0,1\}^*$ to $\mathbb{Z}_p$.*

$\mathsf{KeyGen_S}(gpk, gsk)$ *Choose $x \in_R \mathbb{Z}_p$, and compute $A = g^{\frac{1}{x+\gamma}}$. Output $spk = \emptyset$ and $ssk = (A, x)$.*

$\mathsf{KeyGen_V}(1^\kappa)$**:** *Choose $x_v, y_v, z_v, r_v \in_R \mathbb{Z}_p$, and compute $h_d = g^{x_v y_v r_v}$, $u_d = g^{y_v r_v}$, $v_d = g^{x_v r_v}$, and $t_d = v^{z_v}$. Output $vpk = (h_d, u_d, v_d, t_d)$ and $vsk = (x_v, y_v, z_v)$.*

$\mathsf{Sign}(gpk, ssk, vpk, M)$**:** *Choose $a, b, \alpha, \beta, \delta \in_R \mathbb{Z}_p$, and compute $T_1 = A \cdot h^{\alpha+\beta}$, $T_2 = u^\alpha$, $T_3 = v^\beta$, $D_1 = T_1 \cdot h_d^{a+b}$, $D_2 = u_d^a$, $D_3 = v_d^b$, $S_1 = f^{x_i+\delta}$, and $S_2 = t_d^\delta$. Let $\tau = \alpha x$ and $\lambda = \beta x$. Compute SPK as follows:*

- *Choose $r_x, r_\alpha, r_\beta, r_\delta, r_\tau, r_\lambda, s_{z_v}, c_v \in_R \mathbb{Z}_p$.*
- *Compute $R_v = v^{s_{z_v}} t_d^{-c_v}$, $R_{s,1} = u^{r_\alpha}$, $R_{s,2} = v^{r_\beta}$, $R_{s,3} = e(T_1, g)^{r_x} \cdot e(h, \omega)^{-r_\alpha - r_\beta} \cdot e(h, g)^{-r_\tau - r_\lambda}$, $R_{s,4} = T_2^{r_x} \cdot u^{-r_\tau}$, $R_{s,5} = T_3^{r_x} \cdot v^{-r_\lambda}$, $R_{s,6} = f^{r_x + r_\delta}$, and $R_{s,7} = t_d^{r_\delta}$. Compute $c = H(T_1, T_2, T_3, D_1, D_2, D_3, S_1, S_2, R_v, R_{s,1}, \ldots, R_{s,7}, M)$, $c_s = c - c_v \bmod p$, $s_x = r_x + c_s x$, $s_\alpha = r_\alpha + c_s \alpha$, $s_\beta = r_\beta + c_s \beta$, $s_\delta = r_\delta + c_s \delta$, $s_\tau = r_\tau + c_s \tau$, and $s_\lambda = r_\lambda + c_s \lambda$.*
- *Output $\sigma = (T_2, T_3, D_1, D_2, D_3, S_1, S_2, c_s, c_v, s_x, s_\alpha, s_\beta, s_\delta, s_\tau, s_\lambda, s_{z_v})$.*

$\mathsf{Revoke}(gpk, gsk, ssk, RL)$**:** *Let $ssk = (A, x)$. Compute v^x and $Cert_{A,x} = \mathsf{DSig}(sigkey, v^x)$. Output the updated list $RL \cup (v^x, Cert_{A,x})$.*

$\mathsf{Verify}(gpk, vsk, M, \sigma, RL)$**:** *Output 1 if both the following verification check and revocation check algorithms output 1, and output 0, otherwise.*

Verification check: *Compute $T_1' = D_1/(D_2^{x_v} D_3^{y_v})$, $R_v' = v^{s_{z_v}} t_d^{-c_v}$, $R_{s,1}' = u^{s_\alpha} T_2^{-c_s}$, $R_{s,2}' = v^{s_\beta} T_3^{-c_s}$, $R_{s,3}' = e(T_1', g)^{s_x} \cdot e(h, \omega)^{-s_\alpha - s_\beta} \cdot e(h, g)^{-s_\tau - s_\lambda} (\frac{e(T_1', \omega)}{e(g,g)})^{c_s}$, $R_{s,4}' = T_2^{s_x} \cdot u^{-s_\tau}$, $R_{s,5}' = T_3^{s_x} \cdot v^{-s_\lambda}$, $R_{s,6}' = g^{s_x + s_\delta} S_1^{-c_s}$, and $R_{s,7}' = t_d^{s_\delta} S_2^{-c_s}$. Output 1, if $c_s + c_v = H(T_1', T_2, T_3, D_1, D_2, D_3, S_1, S_2, R_v', R_{s,1}', \ldots, R_{s,7}', M)$ holds, and output 0, otherwise.*

Revocation check: *For all $(v^x, Cert_{A,x}) \in RL$, verify $Cert_{A,x}$ by using verkey, and check $e(S_1, t_d) \stackrel{?}{=} e((v^x)^{z_v} S_2, f)$. If there exists a pair $(v^x, Cert_{A,x}) \in RL$, where $Cert_{A,x}$ is a valid certificate and the above condition holds, then output 1. Otherwise, output 0.*

Note that $e(S_1, t_d) = e(f^{x+\delta}, v^{z_v}) = e(f, v)^{z_v(x+\delta)}$ and $e((v^x)^{z_v} S_2, f) = e(v^{z_v x} v^{z_v \delta}, f) = e(v, f)^{z_v(x+\delta)}$ hold, and $e((v^x)^{z_v} S_2, f)$ can only be computed by the designated verifier (who has z_v).

Next, we describe the simulated signing algorithm as follows:

Protocol 2. *The simulated signing algorithm*

$\mathsf{Sign}'(gpk, spk, vsk, M)$**:** *Choose* $T_2, T_3, D_1, D_2, D_3, S_1, S_2 \in_R \mathbb{G}$*. Compute SPK as follows:*

– *Choose* $s_x, s_\alpha, s_\beta, s_\delta, s_\tau, s_\lambda, r_{z_v}, c_s \in_R \mathbb{Z}_p$.
– *Compute* $R_v = v^{r_{z_v}}$, $R_{s,1} = u^{s_\alpha} T_2^{-c_s}$, $R_{s,2} = v^{s_\beta} T_3^{-c_s}$, $R_{s,3} = e(T_1, g)^{s_x} \cdot e(h, \omega)^{-s_\alpha - s_\beta} \cdot e(h, g)^{-s_\tau - s_\lambda} (\frac{e(T_1, \omega)}{e(g,g)})^{c_s}$, $R_{s,4} = T_2^{s_x} \cdot u^{-s_\tau}$, $R_{s,5} = T_3^{s_x} \cdot v^{-s_\lambda}$, $R_{s,6} = g^{s_x + s_\delta} S_1^{-c_s}$, *and* $R_{s,7} = t_d^{s_\delta} S_2^{-c_s}$. *Compute* $c = H(T_1, T_2, T_3, D_1, D_2, D_3, S_1, S_2, R_v, R_{s,1}, \ldots, R_{s,7}, M)$, $c_v = c - c_s \bmod p$, *and* $s_{z_v} = r_{z_v} + c_v z_v$.
– *Output* $\sigma = (T_2, T_3, D_1, D_2, D_3, S_1, S_2, c_s, c_v, s_x, s_\alpha, s_\beta, s_\delta, s_\tau, s_\lambda, s_{z_v})$.

Obviously, a signature generated by the Sign' algorithm is a valid signature. Therefore, our ADVS scheme satisfies Non-transferability.

Can *RL* be publicly opened?: In our scheme, RL is used to execute the Verify algorithm. Therefore, RL is given to verifiers only. Even if RL is given to a third party, the third party cannot execute the revocation check. However, a different problem occurs. If RL is publicly opened, then the third party can obtain the number of revoked signers. To prevent this, in a natural way, dummy certificates can be used as follows: Let N be the number of group members. Then GM chooses $v'_i \in_R \mathbb{G}$, where $i = 1, 2, \ldots, N - |RL|$. Note that this procedure can deal with a dynamic update of RL, namely, dummy certificates are chosen for each revocation. Although the cost of revocation check and updating the list are increased, ***RL*** can be opened. However, as with VLR schemes, a signer does not need RL to make a signature. Therefore, practically, we can assume that RL is given to verifiers only. In this setting, we can prevent a revoked user from making a valid signature that is verified by the third party, since the third party cannot verify the validity of a signature by using only gpk. However, in VLR schemes, the third party can verify the validity of a signature by using gpk only, since RL is used for the revocation check only. Therefore, VLR group signature schemes are not used (under the assumption that RL is given to verifiers only), since a revoked user could make a valid group signature which could be verified by the third party. This is a superior point of our scheme compared with VLR schemes.

The Open algorithm: The Open algorithm is described as follows: $A \leftarrow \mathsf{Open}(gpk, gsk, (M, \sigma))$, where A is a signer secret key. Let $\xi_1 := \log_u h$ and $\xi_2 := \log_v h$. By adding (ξ_1, ξ_2) to gsk, GM can compute $T_1/(T_2^{\xi_1} T_3^{\xi_2})$ if T_1 is given. Therefore, the designated verifier needs to send (T'_1, T_2, T_3) to GM to request the Open procedure. If the opening and issuing roles need to be separated, then only the opening key $osk = (\xi_1, \xi_2)$ is given to the Opening Manager OM. A designated verifier sends (T'_1, T_2, T_3) to OM. If (T'_1, T_2, T_3) is included in a signature computed by the simulated signing algorithm Sign', then the Open algorithm does not work, since (T'_1, T_2, T_3) is not a valid ciphertext of a membership certificate A (T_2 and T_3 are randomly chosen). Therefore, Non-transferability is not satisfied from the viewpoint of OM. This suggests OM can reveal not only the identity of a signer, but also information about who the actual signer is.

5 Security Analysis

In this section, we prove that our scheme satisfies security requirements defined in Section 3.

Theorem 1. *Our scheme satisfies Unforgeability under the q-SDH assumption.*

Proof. Let $\mathcal{A}$ be an adversary to break Unforgeability of our scheme. We construct an algorithm $\mathcal{B}$ to break the q-SDH problem: Let $(g_1, g_1^{\gamma}, \cdots, g_1^{\gamma^q})$ be an instance of q-SDH problem. Let q_n be the number of signers ($q_n \leq q$). W.l.o.g., we assume that $q_n = q$. $\mathcal{B}$ chooses distinct $x_1, \ldots, x_{q-1} \in_R \mathbb{Z}_p$, and sets $f(X) := \prod_{i=1}^{q-1}(X + x_i) := \sum_{i=0}^{q-1} \alpha_i X^i$, where $\alpha_0, \ldots, \alpha_{q-1} \in \mathbb{Z}_p$ are the coefficients of the polynomial f. $\mathcal{B}$ chooses $\theta \in_R \mathbb{Z}_p$, and computes $g' := \prod_{i=0}^{q-1}(g_1^{\gamma^i})^{\alpha_i \theta} = g_1^{\theta f(\gamma)}$ and $g'' := \prod_{i=1}^{q}(g_1^{\gamma^i})^{\alpha_{i-1}\theta} = g_1^{\theta\gamma f(\gamma)} = (g')^{\gamma}$. Let $f_i(X) := f(X)/(\gamma + x_i) = \prod_{j=1, j\neq i}^{q-1}(X + x_j) := \sum_{j=0}^{q-2} \beta_i X^j$, where $\beta_0, \ldots, \beta_{q-2} \in \mathbb{Z}_p$ are the coefficients of the polynomial f_i. Then $A_i = \prod_{j=0}^{q-2}(g_1^{\gamma^j})^{\beta_j \theta} = g_1^{\theta f_i(\gamma)} = (g')^{1/(\gamma + x_i)}$ is a signer public key. $\mathcal{B}$ sets $g := g'$ and $\omega := g'' = g^{\gamma}$. $\mathcal{B}$ chooses $h, u, v, f \in_R \mathbb{G}$, $x_v, y_v, z_v, r_v \in_R \mathbb{Z}_p$, and computes $h_d = g^{x_v y_v r_v}$, $u_d = g^{y_v r_v}$, $v_d = g^{x_v r_v}$, and $t_d = v^{z_v}$. $\mathcal{B}$ gives $gpk = (e, (\mathbb{G}, \mathbb{G}_T), g, h, u, v, f, \omega, H)$ and $vpk = (h_d, u_d, v_d, t_d)$ to $\mathcal{A}$, where $H : \{0,1\}^* \to \mathbb{Z}_p$ is a random oracle. In addition, $\mathcal{B}$ selects a signing key of DSig $sigkey$, and opens a corresponding verification key $verkey$. For verification queries and signing queries issued by $\mathcal{A}$, $\mathcal{B}$ can answer these queries perfectly, since $\mathcal{B}$ has $vsk = (x_v, y_v, z_v)$, and can execute the simulated signing algorithm Sign'. For a corruption query i, $\mathcal{B}$ returns (A_i, x_i) to $\mathcal{A}$. For a revocation query i, $\mathcal{B}$ computes v^{x_i} and $Cert_{A_i, x_i} = \mathsf{DSig}(sigkey, v^{x_i})$, and outputs updated list $RL \cup (v^{x_i}, Cert_{A_i, x_i})$. $\mathcal{A}$ outputs (M^*, σ^*). Let $\sigma^* = (T_2, T_3, D_1, D_2, D_3, S_1, S_2, c_s, c_v, s_x, s_\alpha, s_\beta, s_\delta, s_\tau, s_\lambda, s_{z_v})$. $\mathcal{B}$ computes $T_1 = D_1/(D_2^{x_v} D_3^{y_v})$, and can obtain $(T_1, T_2, T_3, c_s, s_\alpha, s_\beta, s_\tau, s_\lambda)$. By using the Forking Lemma [19], $\mathcal{B}$ can obtain $(T_1, T_2, T_3, c_s', s_\alpha', s_\beta', s_\tau', s_\lambda')$, where $c_s \neq c_s'$, with non-negligible probability. By using Lemma 4.4 of [3], we can extract a new SDH tuple $(\tilde{A}, \tilde{x})$ as follows: Let $\Delta c_s := c_s - c_s'$, $\Delta s_\alpha := s_\alpha - s_\alpha'$, $\Delta s_\beta := s_\beta - s_\beta'$, $\Delta s_x := s_x - s_x'$, $\Delta s_\tau := s_{-} s_\tau'$, $\Delta s_\lambda := s_\lambda - s_\lambda'$, $\tilde{\alpha} := \Delta s_\alpha / \Delta c_s$, $\tilde{\beta} := \Delta s_\beta / \Delta c_s$, $\tilde{x} := \Delta s_x / \Delta c_s$, and $\tilde{A} := T_1 \cdot h^{-\tilde{\alpha} - \tilde{\beta}}$. Therefore, $\mathcal{B}$ can solve q-SDH problem. □

Theorem 2. *Our scheme satisfies Signer anonymity under the DLIN assumption in the random oracle model.*

To prove Theorem 2, we apply the BBS short group signature scheme and CPA-full anonymity experiment. For the sake of clarity, we introduce the BBS scheme and the definition of CPA-full anonymity in Appendices A.1 and A.2, respectively.

Proof. Let $\mathcal{A}$ be an adversary to break Signer anonymity of our scheme. We construct an algorithm $\mathcal{B}$ to break CPA-full-anonymity of the BBS short group signature scheme with 2-person group as follows: First, the challenger $\mathcal{C}$ sends $(e, (\mathbb{G}, \mathbb{G}_T), g, \omega, H)$, ssk_0, and ssk_1 to $\mathcal{B}$. $\mathcal{B}$ chooses $h, u, v, f \in_R \mathbb{G}$, $x_v, y_v, z_v, r_v \in_R \mathbb{Z}_p$, and computes $h_d = g^{x_v y_v r_v}$, $u_d = g^{y_v r_v}$, $v_d = g^{x_v r_v}$, and $t_d = v^{z_v}$. $\mathcal{B}$ gives $gpk = (e, (\mathbb{G}, \mathbb{G}_T), g, h, u, v, f, \omega, H)$, $vpk = (h_d, u_d, v_d, t_d)$, $vsk = (x_v, y_v, z_v)$, ssk_0, and ssk_1, where $H : \{0,1\}^* \to \mathbb{Z}_p$ is a hash function. In addition, $\mathcal{B}$ selects a signing key of DSig *sigkey*, and opens a corresponding verification key *verkey*. $\mathcal{A}$ sends M^* to $\mathcal{B}$. $\mathcal{B}$ forwards M^* to $\mathcal{C}$, and obtains $\sigma^* = (T_1, T_2, T_3, c_s, s_x, s_\alpha, s_\beta, s_\tau, s_\lambda)$. $\mathcal{B}$ chooses $s_\delta, r_{z_v}, c_v \in_R \mathbb{Z}_p$ and $S_1, S_2 \in_R \mathbb{G}$. $\mathcal{B}$ computes $R_v = v^{r_{z_v}} t_d^{-c_v}$, $R_{s,1} = u^{s_\alpha} T_2^{-c_s}$, $R_{s,2} = v^{s_\beta} T_3^{-c_s}$, $R_{s,3} = e(T_1, g)^{s_x} \cdot e(h, \omega)^{-s_\alpha - s_\beta} \cdot e(h, g)^{-s_\tau - s_\lambda} (\frac{e(T_1,\omega)}{e(g,g)})^{c_s}$, $R_{s,4} = T_2^{s_x} \cdot u^{-s_\tau}$, $R_{s,5} = T_3^{s_x} \cdot v^{-s_\lambda}$, $R_{s,6} = g^{s_x + s_\delta} S_1^{-c_s}$, and $R_{s,7} = t_d^{s_\delta} S_2^{-c_s}$. $\mathcal{B}$ also computes $s_{z_v} = r_{z_v} + c_v z_v$, and sets $c := H(T_1, T_2, T_3, D_1, D_2, D_3, S_1, S_2, R_v, R_{s,1}, \ldots, R_{s,7}, M^*)$, where $c = c_v + c_s \bmod p$. $\mathcal{B}$ sends the challenge signature $(T_2, T_3, D_1, D_2, D_3, S_1, S_2, c_s, c_v, s_x, s_\alpha, s_\beta, s_\delta, s_\tau, s_\lambda, s_{z_v})$ to $\mathcal{A}$. $\mathcal{A}$ outputs μ'. Finally, $\mathcal{B}$ outputs μ' as the answer to the anonymity game of the BBS group signature scheme. Therefore, our scheme satisfies Signer anonymity under the DLIN assumption, since the BBS group signature scheme satisfies anonymity under the DLIN assumption in the random oracle model. □

The following theorem clearly holds, since there exists the simulated signing algorithm Sign', and OM with a linear encryption secret key (ξ_1, ξ_2) can reveal information about who the actual signer is.

Theorem 3. *Our scheme satisfies Non-transferability under the DLIN assumption.*

6 Applications of our ADVS scheme

In this section, we show the applications of our scheme to a biometric-based remote authentication scheme (the BCPZ scheme [5]) and an identity management scheme (the IMSTY scheme [11]).

6.1 Biometric Authentication

The BCPZ scheme [5] is based on the Boneh and Shacham VLR group signature [4]. $\mathcal{H}$ is a human user (who authenticates himself/herself to a service provider $\mathcal{P}$ by using his/her biometric data b preserved on a plastic card). A sensor client $\mathcal{S}$ extracts human user's biometric trait (e.g., iris is used in the BCPZ scheme), and communicates with $\mathcal{P}$, so that the user will be authenticated by $\mathcal{P}$. $\mathcal{P}$ executes KeyGen_V, and obtains vpk and vsk. A card issuer $\mathcal{I}$ (with a group secret key γ) issues a card to a human user, and $(A = g^{\frac{1}{x+\gamma}}, b)$ is preserved in the card, where b is biometric data of the user and $x = Hash(b)$. In addition, $\mathcal{I}$ generates RL if malicious behavior occurs or a user loses his/her cards. First, $\mathcal{P}$ sends the challenge M to $\mathcal{S}$. $\mathcal{S}$ gets (A, b) and the *fresh* biometric trait b' from a human user (with a card), confirms $b' \sim b$ (which indicates that b' and b are acquired from the same biometric source), and computes $x = Hash(b)$ and a group signature σ by using a secret x and vpk. $\mathcal{P}$ verifies (M, σ), and checks whether the user is a malicious user or not, by using RL. In the (original) BCPZ scheme, a third party (with RL) may think that:

- There might be many malicious behaviors in this company.
- There might be many lost cards, i.e., goods management may deteriorate in this company.

and so on. This is where our ADVS scheme comes into effect. We illustrate a modified BCPZ scheme in Fig.1.

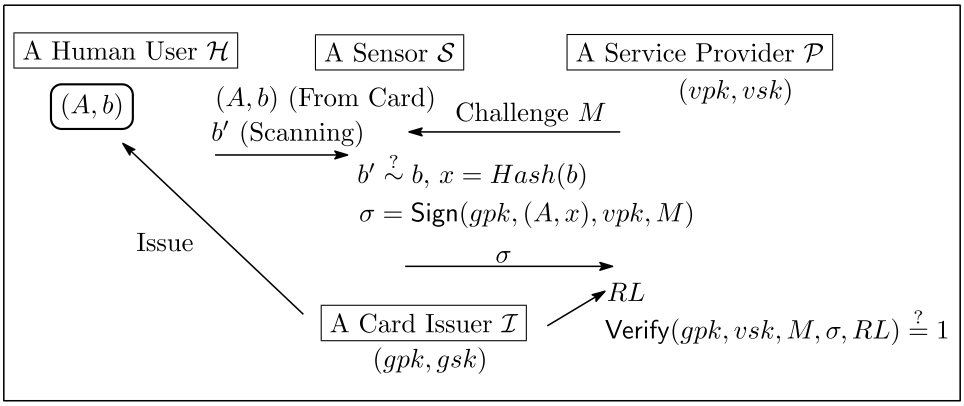

Fig. 1: Modified BCPZ scheme

We assume that RL is given to $\mathcal{P}$ only, or that RL is opened with dummy certificates. The service provider $\mathcal{P}$ does not have to manage the identity of each user. Users do not have to manage any extra values (e.g., passwords), since they only use their own biometric traits and their cards.

6.2 Identity management

An outsourcing business using group signature has been proposed in [11] (called the IMSTY scheme). In existing systems (which do not apply group signature), authentication servers store the list of identities of users. In group signature settings, authentication servers only have to verify users by using the group public key gpk, and do not have to manage the list of identities of users $ID\text{-}list$. Therefore, the risk of leaking user information (i.e., the list of identities of users) can be minimized, and this is the merit of using group signature in identity management. In the IMSTY scheme, the role of Group Manager GM is separated into three roles: Issuing Manager IM, User-Revocation Manager RM, and Opening Manager OM. IM issues membership certificates for users. When a user requests the service, the user makes a group signature σ, and sends it to Outsourcee who is in charge of providing the service to legitimate users. Outsourcee verifies σ, provides the service if this signature is valid, and stores σ into the usage log $ULog$. After a certain interval, Outsourcee sends $ULog$ to OM who can open group signatures. OM charges the users who have already used the service. If a user does not pay a fee, then OM announces the identity of this user to RM. RM updates the revocation list RL when a user wants to leave the group, or when a user does not pay a fee. *ID-list* is managed by IM, and it is updated when a new user joins. IM sends $ID\text{-}list = \{(A, x), UserID\}$ to OM, namely Outsourcee does not have to manage *ID-list*. In the (original) IMSTY scheme, a third party may think that:

– There might be many seceders, i.e., this service may not be interesting.
– Signer's rights have been revoked, maybe, he/she did not pay the service fee. That is to say, the service fee may be expensive.

and so on. This is where our ADVS scheme comes into effect. GM of our ADVS scheme also can be separated into three roles, since γ (which is used to issue membership certificates) is not used for executing the Revoke algorithm, and the Open algorithm is independent of other procedures. We illustrate a modified IMSTY scheme in Fig.2.

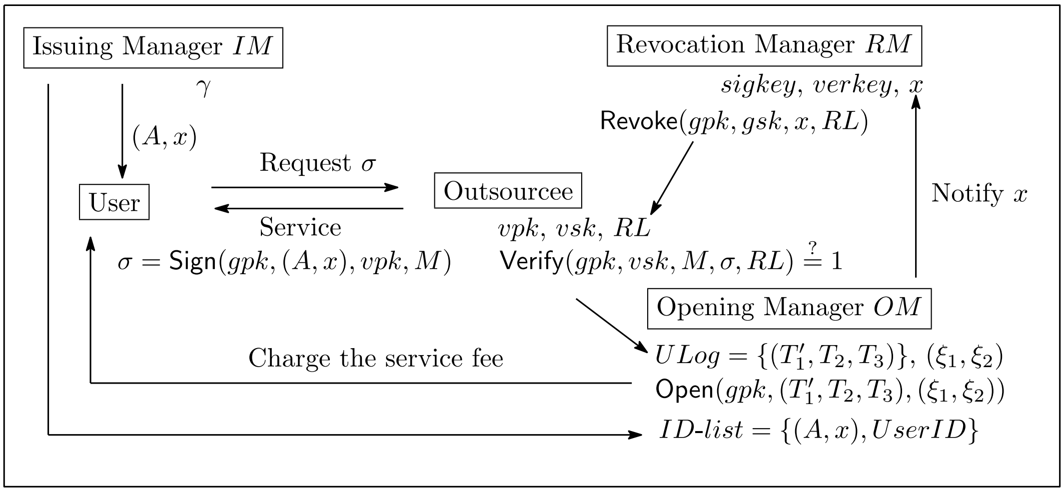

Fig. 2: Modified IMSTY scheme

We assume that RL is given to Outsourcee only, or that RL is opened with dummy certificates, and all entities know the group public key gpk. In the modified IMSTY scheme, (T_1', T_2, T_3) is stored into $ULog$, since the signature validity has already been checked by Outsourcee, and OM needs (T_1', T_2, T_3) only to execute the Open procedure. After a certain interval, Outsourcee sends $ULog$ to OM, and OM charges the users who have already used the service. If a user does not pay a fee, then OM notifies x of this user to RM. RM updates the revocation list RL, and sends it to Outsourcee, or opens RL with dummy certificates $v_i' \in \mathbb{G}$ $(i = 1, 2, \ldots, N - |RL|)$, where N is the number of group members.

7 Conclusion

In this paper, we propose an ADVS scheme with revocation. Our ADVS scheme satisfies not only designated verification and Signer anonymity, but also designated revocation check. To the best of our knowledge, our scheme is the first provably secure scheme with designated revocation check. Our scheme can be applied to the *protecting company's reputation* scenario. Neither strong DVS nor revocable group signature schemes can be used in this situation. Our ADVS scheme can be directly and easily applied to the BCPZ scheme and the IMSTY scheme. From this fact, our ADVS scheme can be directly and easily applied to many cryptographic schemes based on (revocable) group signatures, when designated property is required.

Acknowledgements

The authors would like to thank anonymous reviewers of ProvSec 2010 for their invaluable comments. The first author Keita Emura is supported by the Center for Highly Dependable Embedded Systems Technology as a Postdoc researcher.

References

1. Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *EUROCRYPT*, pages 614–629, 2003.
2. Dan Boneh and Xavier Boyen. Short signatures without random oracles and the SDH assumption in bilinear groups. *J. Cryptology*, 21(2):149–177, 2008.
3. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *CRYPTO*, pages 41–55, 2004.
4. Dan Boneh and Hovav Shacham. Group signatures with verifier-local revocation. In *ACM Conference on Computer and Communications Security*, pages 168–177, 2004.
5. Julien Bringer, Hervé Chabanne, David Pointcheval, and Sébastien Zimmer. An application of the Boneh and Shacham group signature scheme to biometric authentication. In *IWSEC*, pages 219–230, 2008.
6. David Chaum and Eugène van Heyst. Group signatures. In *EUROCRYPT*, pages 257–265, 1991.
7. Sherman S. M. Chow and Duncan S. Wong. Anonymous identification and designated-verifiers signatures from insecure batch verification. In *EuroPKI*, pages 203–219, 2007.
8. Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO*, pages 174–187, 1994.
9. Keita Emura, Atsuko Miyaji, and Kazumasa Omote. A certificate revocable anonymous authentication scheme with designated verifier. In *ARES*, pages 769–773, 2009.
10. Xinyi Huang, Willy Susilo, Yi Mu, and Futai Zhang. Short (identity-based) strong designated verifier signature schemes. In *ISPEC*, pages 214–225, 2006.

11. Toshiyuki Isshiki, Kengo Mori, Kazue Sako, Isamu Teranishi, and Shoko Yonezawa. Using group signatures for identity management and its implementation. In *Digital Identity Management*, pages 73–78, 2006.
12. Markus Jakobsson, Kazue Sako, and Russell Impagliazzo. Designated verifier proofs and their applications. In *EUROCRYPT*, pages 143–154. Springer-Verlag, 1996. LNCS 1070.
13. Fabien Laguillaumie and Damien Vergnaud. Designated verifier signatures: Anonymity and efficient construction from any bilinear map. In *SCN*, pages 105–119, 2004.
14. Fabien Laguillaumie and Damien Vergnaud. Multi-designated verifiers signatures: anonymity without encryption. *Inf. Process. Lett.*, 102(2-3):127–132, 2007.
15. Benoit Libert and Damien Vergnaud. Group signatures with verifier-local revocation and backward unlinkability in the standard model. In *CANS*, pages 498–517, 2009.
16. Chunbo Ma and Jianhua Li. Adaptable designated group signature. In *ICIC (1)*, pages 1053–1061, 2006.
17. Toru Nakanishi, Hiroki Fujii, Yuta Hira, and Nobuo Funabiki. Revocable group signature schemes with constant costs for signing and verifying. In *Public Key Cryptography*, pages 463–480, 2009.
18. Toru Nakanishi and Nobuo Funabiki. A short verifier-local revocation group signature scheme with backward unlinkability. *IEICE Transactions*, 90-A(9):1793–1802, 2007.
19. David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *J. Cryptology*, 13(3):361–396, 2000.
20. Siamak Fayyaz Shahandashti and Reihaneh Safavi-Naini. Construction of universal designated-verifier signatures and identity-based signatures from standard signatures. In *Public Key Cryptography*, pages 121–140, 2008.
21. Ron Steinfeld, Huaxiong Wang, and Josef Pieprzyk. Efficient extension of standard schnorr/RSA signatures into universal designated-verifier signatures. In *Public Key Cryptography*, pages 86–100, 2004.
22. Yaling Zhang, Jing Zhang, and Yikun Zhang. Multi-signers strong designated verifier signature scheme. In *SNPD*, pages 324–328, 2008.

Appendix

A.1 BBS Short Group Signature

In this appendix, we introduce the BBS short group signature [3]. Let $(\mathbb{G}, \mathbb{G}_T)$ be a bilinear group with pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, and $\mathcal{P} = \{U_1, \ldots, U_n\}$ be a set of participants.

Protocol 3. *BBS Short Group Signature [3]*

$\mathsf{KeyGen}(1^\kappa)$**:** *Choose* $g, h \in \mathbb{G}$ *and* $\gamma, \xi_1, \xi_2 \in \mathbb{Z}_p$*, and set* $u = h^{\xi_1}$*,* $v = h^{\xi_2}$*, and* $\omega = g^\gamma$*. For a user* $U_i \in \mathcal{P}$*, choose* $x_i \in_R \mathbb{Z}_p$*, and compute* $A_i = g^{\frac{1}{x_i+\gamma}}$*. Output the group public key* $gpk = (e, (\mathbb{G}, \mathbb{G}_T), g, \omega, H)$*, the group secret key* $gsk = \gamma$*, and user secret keys* $\{ssk_i = (x_i, A_i)\}_{U_i \in \mathcal{P}}$*, where* H *is a cryptographic hash function from* $\{0,1\}^*$ *to* $\mathbb{Z}_p$*.*

$\mathsf{GSig}(gpk, ssk_i, M)$**:** *Choose* $\alpha, \beta, r_x, r_\alpha, r_\beta, r_\tau, r_\lambda \in_R \mathbb{Z}_p$, *and compute* $T_1 = A_i \cdot h^{\alpha+\beta}$, $T_2 = u^\alpha$, $T_3 = v^\beta$, $R_1 = u^{r_\alpha}$, $R_2 = v^{r_\beta}$, $R_3 = e(T_1, g)^{r_x} \cdot e(h, \omega)^{-r_\alpha - r_\beta} \cdot e(h, g)^{-r_\tau - r_\lambda}$, $R_4 = T_2^{r_x} u^{-r_\tau}$, $R_5 = T_3^{r_x} v^{-r_\lambda}$, $c = H(M, T_1, T_2, T_3, R_1, \ldots, R_5)$, $s_x = r_x + c_s x$, $s_\alpha = r_\alpha + c_s \alpha$, $s_\beta = r_\beta + c_s \beta$, $s_\tau = r_\tau + c_s \tau$, *and* $s_\lambda = r_\lambda + c_s \lambda$. *Output* $\sigma = (T_1, T_2, T_3, s_x, s_\alpha, s_\beta, s_\tau, s_\lambda)$.

$\mathsf{GVer}(gpk, \sigma, M)$**:** *Compute* $R_1' = u^{s_\alpha} T_2^{-c}$, $R_2' = v^{s_\beta} T_3^{-c}$, $R_3' = e(T_1, g)^{s_x} \cdot e(h, \omega)^{-s_\alpha - s_\beta} \cdot e(h, g)^{-s_\tau - s_\lambda} \left(\frac{e(T_1, \omega)}{e(g, g)}\right)^c$, $R_4' = T_2^{s_x} u^{-s_\tau}$, *and* $R_5 = T_3^{s_x} v^{-s_\lambda}$, *and check* $c \stackrel{?}{=} H(M, T_1, T_2, T_3, R_1', \ldots, R_5')$. *If checking condition holds, then output 1, and 0, otherwise.*

$\mathsf{Open}(gpk, gsk, \sigma, M)$**:** *Verify that* σ *is a valid signature on* M *to execute* $\mathsf{Verify}(gpk, \sigma, M)$. *Next, compute* $A_i = T_1/(T_2^{\xi_1} T_3^{\xi_2})$, *and return the signer's identity* i.

A.2 CPA-Anonymity

In this appendix, we introduce the definition of full-anonymity [1]. Note that the BBS short group signature is proven under CPA-full-anonymity, where an adversary cannot issue the Open oracle. Therefore, we introduce this weaker security notion as follows:

Definition 7. *CPA-Anonymity*

$$\begin{aligned} Adv_{\mathcal{A}}^{Anon}(\kappa) = \Big| \Pr \big[& (gpk, gsk, \{ssk_i\}_{U_i \in \mathcal{P}}) \leftarrow \mathsf{KeyGen}(1^\kappa); \\ & (M^*, i_0, i_1, State) \leftarrow \mathcal{A}(gpk, \{ssk_i\}_{U_i \in \mathcal{P}}) \\ & \mu \in_R \{0, 1\}; \sigma_\mu \leftarrow \mathsf{GSig}(gpk, ssk_{i_\mu}, M^*); \\ & \mu' \leftarrow \mathcal{A}(\sigma_\mu, State); \mu = \mu' \big] - \frac{1}{2} \Big| \end{aligned}$$

The BBS short signature satisfies CPA-full-anonymity under the DLIN assumption in the random oracle model (Theorem 5.2 of [3]).