A Secure Three-Move Blind Signature Scheme for Polynomially Many Signatures

Masayuki Abe

NTT Laboratories
1-1 Hikari-no-oka, Yokosuka-shi, 239-0847 Japan
abe@isl.ntt.co.jp

Abstract. Known practical blind signature schemes whose security against adaptive and parallel attacks can be proven in the random oracle model either need five data exchanges between the signer and the user or are limited to issue only logarithmically many signatures in terms of a security parameter. This paper presents an efficient blind signature scheme that allows a polynomial number of signatures to be securely issued while only three data exchanges are needed. Its security is proven in the random oracle model. As an application, a provably secure solution for double-spender-traceable e-cash is presented.

1 Introduction

Blind signatures are a key part of some information systems that offer both user privacy and data authenticity. Such systems include anonymous electronic cash and electronic voting as typical examples. The notion of blind signatures was first introduced by Chaum in [12] with the first scheme based on RSA. Later, some discrete-log based signature schemes were turned into blind signatures [24,10,21]. For some applications, extra functionalities, such as partial blindness [2,1,3] and revocability [6,11,9], were added. A secure blind signature scheme should be one-more unforgeable against adaptive and parallel attacks. Namely, users should not be able to produce more signatures than legitimately issued.

There are some theoretical results on the security of blind signatures [14,25,22]. In [22], a formal security definition and a secure scheme were introduced, though the scheme was rather impractical compared to ordinary signature schemes in real use. In [27,29], Pointcheval and Stern proved that one type of efficient blind signature schemes, which includes Okamoto-Schnorr [23] and Okamoto-Guillou-Quisquater [20] signatures, to be secure in the random oracle model [4] as long as a logarithmic number of signatures were issued. Later, [26] introduced a generic adaptation that renders logarithmically secure blind signature schemes into secure ones with polynomially many signatures. Its cost is two additional data transfers. As the underlying schemes require three data transfers, the resulting schemes need five moves of data between the signer and a user. In [30], Schnorr and Jakobsson argued the security of the Schnorr blind signature in the random oracle model with a strong assumption; the attacker is generic, i.e., restricted to use the group operation only. In [17], Fischlin pointed out some pitfalls that

B. Pfitzmann (Ed.): EUROCRYPT 2001, LNCS 2045, pp. 136–151, 2001.

could be found between the generic adversary plus random oracle model and the reality.

This paper presents a blind signature scheme that needs only three data moves and provides polynomial security, i.e., one-more unforgeable even if polynomially many signatures are issued in an adaptive and concurrent manner. The security is proven in the random oracle model. The scheme remains practical as it requires only three to four times more computation than the original Schnorr signatures [31].

Another advantage of our scheme is its potential support of protocols that need additional functionality. By following the idea of [3], one can easily extend our scheme to be partially blind schemes. Furthermore, it is shown that a variant of our scheme gives a provably secure solution for double-spender-traceable electronic cash systems. Note that such e-cash schemes in the literature, e.g. [6,7,18], rely on a variant of blind signatures called restrictive blind signatures [7], whose security has been proved only under non-standard and strong assumptions and only against certain restricted attacks [8] while our solution withstands the most general attacks.

2 Security Definitions

Blind signature schemes have two aspects of security; blindness and one-more unforgeability. Let $(\mathcal{G}, \mathcal{S}, \mathcal{U}, \mathcal{V})$ be a blind signature scheme where $\mathcal{G}$ is the key generation algorithm, $\mathcal{S}$ and $\mathcal{U}$ are a signer and a user, respectively, and $\mathcal{V}$ is a verification algorithm (refer to [22] for a formal definition of blind signature schemes).

Definition 1. *(Blindness) Let $\mathcal{S}^*$ and $\mathcal{D}^*$ be a signer and a distinguisher. Let view_0 and view_1 be views of $\mathcal{S}^*$ during executions of the signature issuing protocol where honest user $\mathcal{U}$ obtains valid signature-message pairs $(\Sigma_0, \mathsf{msg}_0)$ and $(\Sigma_1, \mathsf{msg}_1)$, respectively. Given $(\mathsf{view}_0, \mathsf{view}_1, \Sigma_b, \mathsf{msg}_b)$ for $b \in_U \{0,1\}$, $\mathcal{D}^*$ outputs $b' \in \{0,1\}$. A signature scheme is blind if, for all polynomial-time $\mathcal{S}^*$ and $\mathcal{D}^*$, $b' = b$ happens with probability at most $1/2 + 1/n^c$ for sufficiently large n and some constant c. The probability is taken over the coin flips of $\mathcal{G}$, $\mathcal{S}^*$, $\mathcal{D}^*$ and $\mathcal{U}$.*

Note that our scheme provides *computational* blindness defined as above while some of the previously known schemes achieve *perfect* blindness where the success probability of unbound $\mathcal{D}^*$ is exactly $1/2$.

Definition 2. *(One-more unforgeability) A blind signature scheme is $(\ell, \ell+1)$ unforgeable if, for any probabilistic polynomial-time algorithm $\mathcal{U}^*$, $\mathcal{U}^*$ outputs $\ell+1$ valid signatures with probability at most $1/n^c$ for sufficiently large n and some constant c after interacting with legitimate signer $\mathcal{S}$ at most ℓ times in an adaptive and concurrent manner. The probability is taken over the coin flips of $\mathcal{G}$, $\mathcal{S}$, and $\mathcal{U}^*$.*

In the random oracle model, these success probabilities also depend on the choice of random oracles.

3 The Proposed Scheme

3.1 Underlying Idea

The proposed scheme is based on the partially blind signature scheme of [3]. Roughly, their scheme is a witness indistinguishable variant of the Schnorr signature scheme where the signer uses two public keys $y(=g^x)$ and $z(=g^w)$, which we call the *real public key* and the *tag public key*, respectively, in such a way that the signature can be issued only with real secret key x but no one can distinguish which secret key, i.e., x or w, was used. Their scheme then allows the signer to sign with several different tag public keys to achieve partial blindness. It was proven that the same tag key could be used only for logarithmically many signatures but the signer could use polynomially many tag keys. Accordingly, if the signer generates a one-time tag key each time he signs, it achieves polynomial security, though the blindness is lost.

Our scheme follows the above approach with additional ideas to retain blindness. It allows the user to blind the tag public key so that the resulting signature can be verified with the real public key provided by the signer and the blinded tag public key provided by the user. However, if the blinding is *perfectly* done and the resulting tag public key just looks like a random public key, the user could himself generate such a signature by arbitrarily creating the tag key and exploiting witness indistinguishability. Accordingly, we restrict the blinding so that the resulting blinded tag key maintains a link to the original one but the link is computationally hidden. Namely, our scheme provides *computational* blindness. The main idea to realize this property is to use a pair of tag public-keys, say (z, z_1), in such a way that z is fixed and z_1 is changed for every signature. The user blinds them into $(\zeta, \zeta_1) = (z^\gamma, z_1^\gamma)$ with random factor γ so that $\log_z z_1 = \log_\zeta \zeta_1$ holds. Accordingly, (ζ, ζ_1) preserves the relation that underlies (z, z_1). The blindness is now provided if the signer cannot decide whether (z, z_1, ζ, ζ_1) is in such relation or not. Some more tricks are added to force the user follow the blinding procedure to get valid signatures.

This restrictive blinding stealthily preserves the link between each valid signature to a particular execution of the issuing protocol. Thus, if $\ell + 1$ signatures are generated after ℓ executions of the signing protocol, there exists an execution that yields at least two signatures. Accordingly, we only need to consider the possibility of yielding two signatures from one issuing, which results in more efficient reduction than the previous results.

3.2 Construction

Let $\mathcal{G}$ be a probabilistic polynomial-time algorithm that takes security parameter n and outputs (p, q, g) where p, q are large primes that satisfy $q|p-1$, and g is an element of $\mathbb{Z}_p^*$ whose order is q. By $\langle g \rangle$, we denote a prime subgroup in $\mathbb{Z}_p^*$ generated by g. Let $\mathcal{H}_1 : \{0,1\}^* \to \langle g \rangle$, $\mathcal{H}_2 : \{0,1\}^* \to \langle g \rangle$, and $\mathcal{H}_3 : \{0,1\}^* \to \mathbb{Z}_q$ be hash functions. We assume that it is hard to compute the discrete log of the outputs of $\mathcal{H}_1$ and $\mathcal{H}_2$. Such hash functions may be constructed in practice as $\mathrm{SHA}(\mathsf{str})^{(p-1)/q} \bmod p$ allowing negligibly small error probability [3].

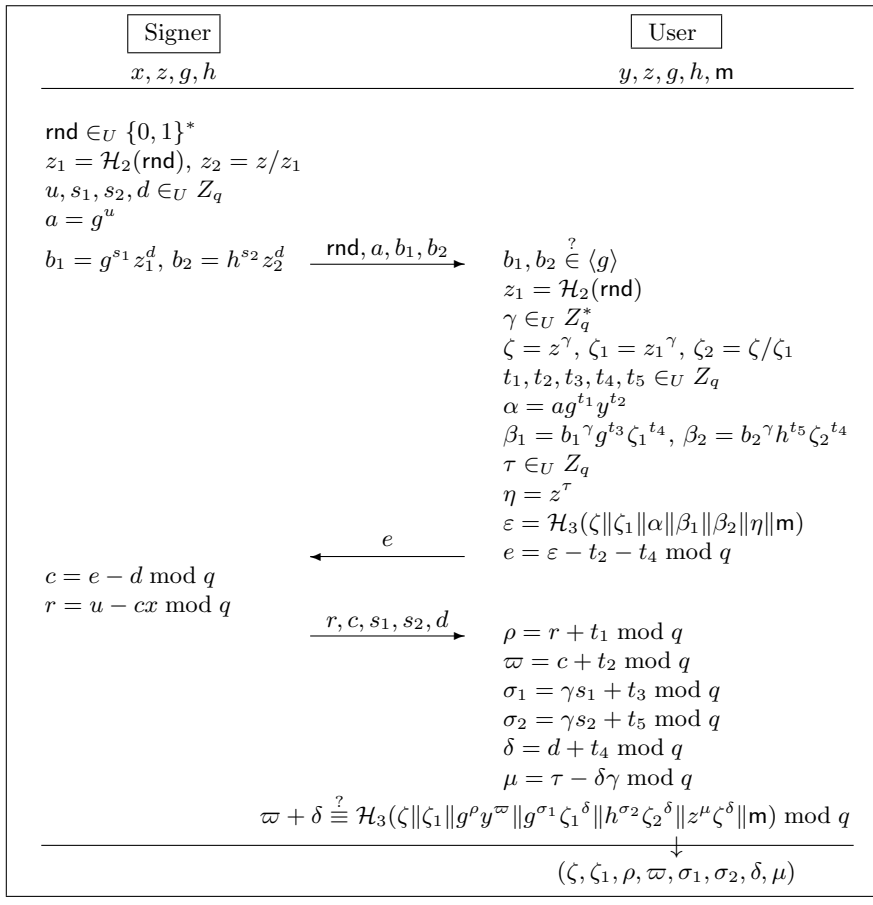

Fig. 1. The signature issuing protocol. The user aborts if any of the checks ($\stackrel{?}{\equiv}$, $\stackrel{?}{\in}$) fails.

[Key Generation]
The signer executes $(p, q, g) \leftarrow \mathcal{G}(1^n)$, and selects $h \in_U \langle g \rangle$, $x \in_U \mathbb{Z}_q$. It then computes real public-key y and fixed tag key z as $y = g^x \bmod p$ and $z = \mathcal{H}_1(p \| q \| g \| h \| y)$, respectively. If $z = 1$, abandon the key and retry. The public key is (p, q, g, h, y, z), and the private key is x.

[Signature Issuing]
Here we overview the signature issuing protocol at a higher level. The details are illustrated in Figure 1. Hereafter, all arithmetic operations are done in $\mathbb{Z}_p$ unless otherwise noted.

Signer $\mathcal{S}$: $\mathcal{S}$ generates a random string rnd and a one-time tag key $z_1 = \mathcal{H}_2(\mathsf{rnd})$. Sending rnd convinces $\mathcal{U}$ that $\log_g z_1$ is not known to $\mathcal{S}$. Then z_2 is computed so that $z = z_1 \cdot z_2$ holds. The rest of the issuing protocol consists of two parts:

– **y-side:** Proof of knowledge x of $y = g^x$, and
– **z-side:** Proof of knowledge (w_1, w_2) of $z_1 = g^{w_1}$, $z_2 = h^{w_2}$.

Since z-side witness is not known to $\mathcal{S}$, the z-side proof is done by simulation as illustrated in Figure 1 by using the OR-proof technique of [13]. Accordingly, $\mathcal{S}$ can complete the protocol only with y-side witness x.

User $\mathcal{U}$: $\mathcal{U}$ blinds and converts the y-side proof into a signature in the same way as done in Schnorr blind signatures [24,10]. For z-side, $\mathcal{U}$ blinds z, z_1, z_2 into ζ, ζ_1, ζ_2 by raising them with random factor γ. The proofs for z_1, z_2 given from $\mathcal{S}$ are also blinded, and then converted into signatures in the standard way with adjustment for the effect of γ. $\mathcal{U}$ then creates an additional Schnorr signature that proves $\zeta = z^\gamma$.

The resulting signature Σ is 8-tuple $\Sigma = (\zeta, \zeta_1, \rho, \varpi, \sigma_1, \sigma_2, \delta, \mu)$ that proves the knowledge of $\log_g y \vee (\log_g \zeta_1 \wedge \log_h(\zeta/\zeta_1) \wedge \log_z \zeta)$.

[Signature Verification]
A signature message pair (Σ, m) is valid if it satisfies $\zeta \not\equiv 1$ and

$$\varpi + \delta \equiv \mathcal{H}_3(\zeta \| \zeta_1 \| g^\rho y^\varpi \| g^{\sigma_1} {\zeta_1}^\delta \| h^{\sigma_2} (\zeta/\zeta_1)^\delta \| z^\mu \zeta^\delta \| \mathsf{m}) \bmod q.$$

4 Security Proofs

4.1 Correctness

Theorem 1. *If the signer and the user follow the issuing protocol, the resulting signature satisfies the verification predicates with provability 1.*

Proof. Observe that the following holds.

$$\begin{aligned}
\varpi + \delta &= c + t_2 + d + t_4 = e + t_2 + t_4 = \varepsilon \pmod q \\
g^\rho y^\varpi &= g^{r+t_1} y^{c+t_2} = g^{r+cx} g^{t_1} y^{t_2} = a g^{t_1} y^{t_2} = \alpha \\
g^{\sigma_1} \zeta_1^\delta &= g^{\gamma s_1 + t_3} \zeta_1^{d+t_4} = (b_1 z_1^{-d})^\gamma g^{t_3} \zeta_1^{d+t_4} = b_1^\gamma g^{t_3} \zeta_1^{t_4} = \beta_1 \\
h^{\sigma_2} (\zeta/\zeta_1)^\delta &= h^{\gamma s_2 + t_5} \zeta_2^{d+t_4} = (b_2 z_2^{-d})^\gamma h^{t_5} \zeta_2^{d+t_4} = b_2^\gamma h^{t_5} \zeta_2^{t_4} = \beta_2 \\
z^\mu \zeta^\delta &= z^{\tau - \delta\gamma} \zeta^\delta = z^\tau = \eta
\end{aligned}$$

Furthermore, $\zeta \not\equiv 1$ holds as $\gamma \neq 0$ when the user is honest. □

4.2 Blindness

Theorem 2. *The proposed scheme is blind if the decision Diffie-Hellman problem is intractable and H_1, H_2, H_3 are random oracles.*

Proof. (sketch) Suppose that $(\mathcal{S}^*, \mathcal{D}^*)$ is successful in breaking blindness with probability $1/2 + \epsilon$ where ϵ is not negligible. Let t_s be the maximum running time of $\mathcal{D}^*$, which is also polynomially bound. We show that $\mathcal{S}^*$ can be used

to solve the DDH problem. Define $\mathcal{DH} = \{(X_1, X_2, X_3, X_4) \in \langle g \rangle^4 | \log_{X_1} X_2 = \log_{X_3} X_4\}$ and $\mathcal{R} = \{(X_1, X_2, X_3, X_4) \in \langle g \rangle^4\}$. Let $(A, B, C, D) \in \langle g \rangle^4$ be a DDH instance, which is taken from $\mathcal{DH}$ or $\mathcal{R}$ with equal probability. Given such an instance, first define H_1 so that $z = A$. Select $b \in_U \{0, 1\}$ and engage in the issuing protocol with $\mathcal{S}^*$ twice. Label the executions run_0 and run_1. Define H_2 so that $z_1 = B$ in run_b, and $z_1 \in_U \langle g \rangle$ in run_{1-b}. Follow the protocol in both run. Then, generate a signature-message pair (Σ, m) that includes $(\zeta, \zeta_1) = (C, D)$. Other variables in Σ are generated by using the standard zero knowledge simulation technique; randomly choose $\rho, \varpi, \sigma_1, \sigma_2, \delta, \mu$, and then define H_3 so that it looks consistent. Given (Σ, m) and views from $\mathcal{S}$, distinguisher $\mathcal{D}^*$ outputs b'. If $b' = b$, we conclude that the instance is in $\mathcal{DH}$. It is in $\mathcal{R}$, otherwise.

Observe that if $(A, B, C, D) \in \mathcal{DH}$, Σ is a valid signature that can be produced in run_b, since $\log_z z_1 = \log_A B = \log_C D = \log_\zeta \zeta_1$ and there exist blinding factors t_1, t_2, t_3, t_4, t_5 that convert the view of run_b into Σ [1]. On the other hand, Σ cannot be produced from run_{1-b} since $\log_z z_1 \neq \log_\zeta \zeta_1$ except for negligible probability. Therefore, given Σ, $\mathcal{D}^*$ outputs correct b with probability $1/2 + \epsilon$. Next, observe that if $(A, B, C, D) \in \mathcal{R}$, Σ cannot be produced in either run_0 and run_1 since $\log_z z_1 \neq \log_\zeta \zeta_1$ for both runs except for negligible probability. Hence, b is independent of Σ, and $b' = b$ happens with probability $1/2$. Thus, the success probability in DDH problem is $1/2(1/2 + \epsilon) + 1/2(1/2) = 1/2 + \epsilon/2$, which contradicts to the DDH assumption when ϵ is not negligible. Note that $\mathcal{D}^*$ may not terminate in time t_s if the instance is in $\mathcal{R}$. However, this is also to our advantage since we can see that Σ is not a proper input to $\mathcal{D}^*$ and the instance is in $\mathcal{R}$.

Finally, note that if $\mathcal{S}^*$ chooses the same rnd in both executions, the resulting signatures are perfectly indistinguishable as there exist consistent blinding factors for any combination of the views and signatures. □

Note that the blindness relies on the decision Diffie-Hellman assumption over the public key of the signer. This suggests that an adversarial signer could choose p, q, g so that the DDH problem could be solved with those parameters. However, as we shall show in the next section, one-more unforgeability is based on the discrete logarithm assumption. Therefore, choosing weak parameters to violate blindness could result in the loss of one-more unforgeability unless DL is strictly harder than DDH. Nevertheless, it is beneficial for the users to verify that the public keys are generated and the hash functions are chosen so that those assumptions are likely to hold. There are several practical solutions for this matter. An inexpensive solution would be to use a widely believed secure hash function like SHA-1, and plug it into the source of randomness of $\mathcal{G}$ so that the users can believe that there is no room for the adversarial signer to control the resulting parameters. It is also needed to check if y is in $\langle g \rangle$ and z

[1] This is why $b_1, b_2 \stackrel{?}{\in} \langle g \rangle$ has to be checked. Without this check, wrong b_1, b_2 could produce a valid signature if γ is a lucky choice. This results in a nonuniform distribution of γ while the one that underlies the simulated signature follows the uniform distribution.

is correctly made. In practice these could be examined by a certificate authority at registration on behalf of the users.

4.3 One-More Unforgeability

Theorem 3. *The proposed scheme is $(\ell, \ell+1)$-unforgeable for polynomially bound ℓ if the discrete logarithm problem is intractable and H_1, H_2, H_3 are random oracles.*

The proof is structured as follows. We first observe that the scheme is witness indistinguishable [15] (Lemma 1), which helps us to simulate the signer with either y-side or z-side witness(es) to extract the witness of the other side. It is then proven that the user can blind (z, z_1) into (ζ, ζ_1) only in such a way that $\log_z \zeta = \log_{z_1} \zeta_1$ to obtain a valid signature (Lemma 2). We then show that creating a valid signature without engaging in the issuing protocol with the legitimate signer is infeasible (Lemma 3). From Lemma 2 and 3, one can see that if the user engages in the signature issuing protocol ℓ times and outputs $\ell+1$ signatures, there exist at least two valid signatures linked to a particular run of the issuing protocol. So the rest is to prove that such a forger who is successful in producing two signatures from a single protocol run can be used to solve the discrete logarithm problem.

Lemma 1. *The signature issuing protocol is witness indistinguishable.*

The above lemma holds immediately according to [13]. Indeed, it is not hard to see that the issuing protocol can be completed if the signer knows either y-side witness x, or z-side witness $(w_1, w_2) = (\log_g z_1, \log_h z_2)$.

Hereafter, let run_i denote the label of i-th execution of the issuing protocol. We define z-side witness in run_i as (w_{1i}, w_{2i}).

Lemma 2. *(Restrictive Blinding) Let $\mathcal{U}_0^*$ be a user that engages in the signature issuing protocol ℓ times, and outputs a valid message-signature pair, $(\mathsf{m}, \zeta, \zeta_1, \rho, \varpi, \sigma_1, \sigma_2, \delta, \mu)$. Let z_{1i} denote z_1 used by $\mathcal{S}$ in run_i. For polynomially bound ℓ and for all polynomial-time $\mathcal{U}_0^*$, the probability that $\log_z \zeta \neq \log_{z_{1i}} \zeta_1$ holds for all i is negligible if the discrete logarithm problem is intractable and H_1, H_2, H_3 are random oracles.*

Proof idea: Suppose that $\log_g h$ is not known. We assign $z = g^{w_1} h^{w_2}$ and $(z_{1J}, z_{2J}) = (g^{w_1}, h^{w_2})$ for $J \in_U \{1, \ldots, \ell\}$ by defining $\mathcal{H}_1$ and $\mathcal{H}_2$ so. Since the signature contains proofs of $\zeta = z^\gamma$, $\zeta_1 = g^{w_1'}$, $\zeta_2 = h^{w_2'}$, we may be capable of extracting (γ, w_1', w_2') by rewinding the user in the random oracle model. Once it is done, the condition $\log_z \zeta \neq \log_{z_{1J}} \zeta_1$ guarantees that we obtain two different representations of z, i.e., $z = g^{w_1} h^{w_2} = g^{w_1'/\gamma} h^{w_2'/\gamma}$, which allows us to compute $\log_g h$. For this to be done, we need to simulate $\mathcal{S}$ that issues ℓ signatures without knowing $\log_g h$. We do this with y-side witness x by exploiting witness indistinguishability. The problem is that, due to witness indistinguishability, the rewinding may result in extracting y-side witness x, which is already known. So we first flip a coin to decide with which witness, y-side or z-side, the simulation is performed, and expect that one of the following happens.

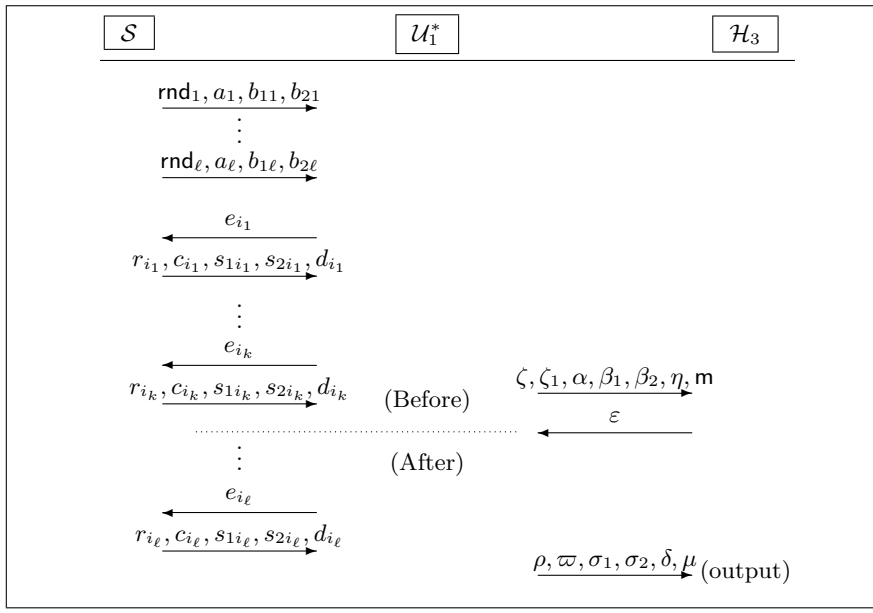

Fig. 2. The interaction among signer $\mathcal{S}$, adversary $\mathcal{U}_1^*$, and random oracle $\mathcal{H}_3$.

- Simulation is done with y-side witness (and z-side witness in run_J). Then another z-side witness is extracted by rewinding. This solves $\log_g h$.
- Simulation is done with z-side witnesses. Then y-side witness is extracted by rewinding. This solves $\log_g y$.

Proof. Assume that, having at most q_h accesses to $\mathcal{H}_3$ and asking at most ℓ signatures to $\mathcal{S}$, $\mathcal{U}_0^*$ outputs signature $(\zeta, \zeta_1, \rho, \varpi, \sigma_1, \sigma_2, \delta, \mu)$ that satisfies $\log_z \zeta_1 \neq \log_{z_{1i}} \zeta_1$ for all i with probability ϵ_0 which is not negligible in n. Here, q_h and ℓ are bound by a polynomial of security parameter n. We randomly fix an index $Q \in \{1, \ldots, q_h\}$ and regard $\mathcal{U}_0^*$ as successful only if the resulting signature corresponds to the Q-th query to $\mathcal{H}_3$. (If it does not correspond to any query, $\mathcal{U}_0^*$ is successful only with negligible probability due to the randomness of $\mathcal{H}_3$.) Accordingly, it is equivalent to assuming an adversary, say $\mathcal{U}_1^*$, that asks $\mathcal{H}_3$ only once and succeeds with probability $\epsilon_1 \geq \epsilon_0/q_h$. Figure 2 illustrates the interaction among the signer $\mathcal{S}$, adversarial user $\mathcal{U}_1^*$, and random oracle $\mathcal{H}_3$. Given $\mathcal{U}_1^*$, we construct machine $\mathcal{M}_1$ that solves the discrete-log problem by simulating the interaction. Let $(\mathbf{p}, \mathbf{q}, \mathbf{g}, \mathbf{Y})$ be an instance to solve $\log_{\mathbf{g}} \mathbf{Y}$ in $\mathbb{Z}_{\mathbf{q}}$.

Reduction Algorithm: $\mathcal{M}_1$ first sets $(p, q, g) := (\mathbf{p}, \mathbf{q}, \mathbf{g})$. It then flips a coin $\chi \in_U \{0, 1\}$ to select either $y := \mathbf{Y}$ (case $\chi = 0$) , or $h := \mathbf{Y}$ (case $\chi = 1$).

Case $y = \mathbf{Y}$: (Extracting y-side witness)

1. $\mathcal{M}_1$ selects $w, w_0 \in_U \mathbb{Z}_q$ and sets $h := g^w$ and $z := \mathcal{H}_1(p \| q \| g \| y) = g^{w_0}$.
2. $\mathcal{M}_1$ runs $\mathcal{U}_1^*$ simulating $\mathcal{S}$ with z-side witnesses as follows.

(a) Select $c_i, r_i \in_U \mathbb{Z}_q$ and compute $a_i := g^{r_i} y^{c_i}$.
(b) Select $\mathsf{rnd}_i \in_U \{0,1\}^*$ and $w_{1i} \in_U \mathbb{Z}_q$ and define $\mathcal{H}_2(\mathsf{rnd}_i)$ as $g^{w_{1i}}$. Then compute $w_{2i} := (w_0 - w_{1i})/w \bmod q$. (Accordingly, $z_{1i} = g^{w_{1i}}$ and $z_{2i} = h^{w_{2i}}$.)
(c) Compute $b_{1i} := g^{u_{1i}}$ and $b_{2i} := h^{u_{2i}}$ with $u_{1i}, u_{2i} \in_U \mathbb{Z}_q$.
(d) Send $\mathsf{rnd}_i, a_i, b_{1i}, b_{2i}$ to $\mathcal{U}_1^*$.
(e) Given e_i from $\mathcal{U}_1^*$, compute $d_i := e_i - c_i \bmod q$, $s_{1i} := u_{1i} - d_i w_{1i} \bmod q$, and $s_{2i} := u_{2i} - d_i w_{2i} \bmod q$.
(f) Send $r_i, c_i, s_{1i}, s_{2i}, d_i$ to $\mathcal{U}_1^*$.

$\mathcal{M}_1$ simulates $\mathcal{H}_3$ by returning $\varepsilon \in_U \mathbb{Z}_q$.

3. $\mathcal{U}_1^*$ outputs a signature, say $(\zeta, \zeta_1, \rho, \varpi, \sigma_1, \sigma_2, \delta, \mu)$, that corresponds to ε.
4. Reset and restart $\mathcal{U}_1^*$ with the same setting. $\mathcal{M}_1$ simulates $\mathcal{H}_3$ with $\varepsilon' \in_U \mathbb{Z}_q$.
5. $\mathcal{U}_1^*$ outputs a signature, say $(\zeta, \zeta_1, \rho', \varpi', \sigma_1', \sigma_2', \delta', \mu')$, that corresponds to ε'.
6. If $\varpi \neq \varpi'$, $\mathcal{M}_1$ outputs $x := (\rho - \rho')/(\varpi' - \varpi) \bmod q$. The simulation fails, otherwise.

Case $h = \mathbf{Y}$: (Extracting z-side witness)

1. $\mathcal{M}_1$ selects $x \in_U \mathbb{Z}_q$ and sets $y := g^x$. It also selects $w_1, w_2 \in_U \mathbb{Z}_q$ and sets $z := \mathcal{H}_1(p \| q \| g \| y) = g^{w_1} h^{w_2}$.
2. $\mathcal{M}_1$ selects $I \in_U \{0, \ldots, \ell\}$ and $J \in_U \{1, \ldots, \ell\}$.
3. $\mathcal{M}_1$ runs $\mathcal{U}_1^*$ simulating as follows.
   (a) For $i \neq J$, $\mathcal{M}_1$ follows the protocol with y-side witness, x. H_2 is simulated by returning random choices from $\langle g \rangle$.
   (b) For $i = J$, $\mathcal{M}_1$ engages in the issuing protocol using both y-side witness x and z-side witness (w_1, w_2) as follows.
      i. Define $\mathcal{H}_2(\mathsf{rnd}_J)$ so that $z_{1J} = g^{w_1}$ and $z_{2J} = h^{w_2}$.
      ii. Compute $a_J = g^{u_J}$, $b_{1J} = g^{u_{1J}}$, $b_{2J} = h^{u_{2J}}$ with $u_J, u_{1J}, u_{2J} \in_U \mathbb{Z}_q$.
      iii. Send $(\mathsf{rnd}_J, a_J, b_{1J}, b_{2J})$ to $\mathcal{U}_1^*$.
      iv. Given e_J from $\mathcal{U}_1^*$, choose $d_J \in_U \mathbb{Z}_q$ and compute $c_J := e_J - d_J \bmod q$, $r_J := u_J - c_J x \bmod q$, $s_{1J} := u_{1J} - d_J w_1 \bmod q$, and $s_{2J} := u_{2J} - d_J w_2 \bmod q$.
      v. Send $(r_J, c_J, s_{1J}, s_{2J}, d_J)$ to $\mathcal{U}_1^*$.

   $\mathcal{M}_1$ simulates $\mathcal{H}_3$ by returning $\varepsilon \in_U \mathbb{Z}_q$.
4. $\mathcal{U}_1^*$ outputs a signature, say $(\zeta, \zeta_1, \rho, \varpi, \sigma_1, \sigma_2, \delta, \mu)$, that corresponds to ε.
5. Rewind and restart $\mathcal{U}_1^*$ with the same setting.
   – If $I = 0$, $\mathcal{M}_1$ simulates $\mathcal{H}_3$ by returning $\varepsilon' \in_U \mathbb{Z}_q$. Otherwise, set $\varepsilon' = \varepsilon$.
   – If $I \neq 0$ and run_J has not yet been completed before the query to $\mathcal{H}_3$ is sent, $\mathcal{M}_1$ simulates the execution by using both y-side and z-side witnesses as above choosing $d_J' \in_U \mathbb{Z}_q$. Otherwise, $\mathcal{M}_1$ simulates only with y-side witness choosing $d_J' = d_J$.
6. $\mathcal{U}_1^*$ outputs a signature, say $(\zeta, \zeta_1, \rho', \varpi', \sigma_1', \sigma_2', \delta', \mu')$, that corresponds to ε'.
7. If $\delta \neq \delta'$, $\mathcal{M}_1$ computes $w_1' = (\sigma_1 - \sigma_1')/(\mu - \mu') \bmod q$, $w_2' = (\sigma_2 - \sigma_2')/(\mu - \mu') \bmod q$, and outputs $w = (w_1 - w_1')/(w_2' - w_2) \bmod q$. Simulation fails, otherwise.

Evaluation of success probability:
In Figure 2, observe that independent variables given to $\mathcal{U}_1^*$ are p, q, g, h, y, $\mathcal{H}_1$, $\mathcal{H}_2$, $\mathsf{rnd}_i, a_i, b_{1i}, b_{2i}$, d_i for all i, and ε and the random tape of $\mathcal{U}_1^*$. All other variables are uniquely determined by these independent variables. Note that e_i's are also determined by the random tape of $\mathcal{U}_1^*$ and the variables that appeared so far. We wrap all these independent variables into Λ, except for $\{\varepsilon, d_{i_{k+1}}, \ldots, d_{i_\ell}\}$, which is defined as D_ε. Let D denote $D_\varepsilon \setminus \{\varepsilon\}$.

Let S be the set of all (Λ, D_ε) that leads $\mathcal{U}_1^*$ to a success, i.e., $\Pr_{\Lambda, D_\varepsilon}[(\Lambda, D_\varepsilon) \in S] \geq \epsilon_1$. According to Lemma 4, with probability at least $\epsilon_1/2$, randomly selected Λ satisfies $\Pr_{D_\varepsilon}[(\Lambda, D_\varepsilon) \in S] \geq \epsilon_1/2$. Once Λ is fixed, δ is uniquely determined by D_ε. By $\delta \leftarrow D_\varepsilon$, we denote the map from (Λ, D_ε) in S to δ. If $(\Lambda, D_\varepsilon) \notin S$, we denote $\perp \leftarrow D_\varepsilon$.

We consider how sensitive δ is to D_ε. Define function ψ as

$$\psi(\delta) = \Pr_{D_\varepsilon}[\delta \leftarrow D_\varepsilon].$$

Let δ_{max} be the value of δ that maximizes $\psi(\delta)$. That is, δ_{max} is the value of δ that is most likely to appear in a successful output of $\mathcal{U}^*$. Let $\psi_{max} = \psi(\delta_{max})$. We consider two cases.

Case 1 (ψ_{max} is not negligible) :
In this case, δ is not likely to change even if D_ε changes, so we perform the rewinding simulation with z-side witnesses choosing D_ε and D'_ε uniformly. By the definition of ψ_{max}, uniformly chosen D_ε and D'_ε yield δ_{max} with probability greater than ψ_{max}^2, which is not negligible. Since ε differs in D_ε and D'_ε with overwhelming probability, we have $\varpi + \delta_{max} = \varepsilon \neq \varepsilon' = \varpi' + \delta_{max} \pmod q$. Thus, we obtain $\varpi \neq \varpi'$ with which y-side witness can be extracted as written in Step-6 of Case $y = \mathbf{Y}$.

Case 2 (ψ_{max} is negligible) :
In this case, δ tends to change if D_ε changes. We first observe that there exists at least one element in D_ε whose change impacts δ. Hereafter, we treat ε in D_ε as d_0, so the elements in D_ε are suffixed as $(0, i_{k+1}, \ldots, i_\ell)$. Define $Id = (0, i_{k+1}, \ldots, i_\ell)$. Let D_ε^{-i} for $i \in Id$ denote a sequence obtained by removing d_i from D_ε. Observe that $\Pr_{D_\varepsilon}[\delta \leftarrow D_\varepsilon] \leq \psi_{max}$ holds for any δ by the definition of ψ_{max}. Suppose that D_ε is uniformly chosen and δ is produced as $\delta \leftarrow D_\varepsilon$. Then, according to Corollary 1, there exists $J \in Id$ such that randomly chosen D_ε^{-J} satisfies

$$\Pr_{d_J}[\delta \leftarrow D_\varepsilon^{-J} \cup \{d_J\}] > 1 - \psi_{max}$$

with probability $< \psi_{max}$. We can correctly guess such index J with probability at least $1/(\ell+1)$ by randomly taking it from $\{0, \ldots, \ell\}$. Taking the complement of the above, we see that randomly chosen D_ε^{-J} satisfies

$$\Pr_{d_J}[\delta \not\leftarrow D_\varepsilon^{-J} \cup \{d_J\}] \geq \psi_{max}$$

with probability $\geq 1 - \psi_{max}$. Now suppose that D'_ε is made from D_ε by choosing $d_J \in_U \mathbb{Z}_q$, and δ' is produced as $\delta' \leftarrow D'_\varepsilon$. From the above observation, $\{\delta' \neq \delta\}$

$\vee\ \{(\Lambda, D'_\varepsilon) \notin S\}$ happens with probability not negligible in n. According to Lemma 4, with probability $\epsilon_1/4$, uniformly chosen D_ε^{-J} satisfies

$$\Pr_{d_J}[(\Lambda, D_\varepsilon^{-J} \cup \{d_J\}) \in S] \geq \epsilon_1/4.$$

Thus, with probability not negligible in n, such D_ε and D'_ε are in S and result in $\delta' \neq \delta$. From this collision, z-side witness $\log_g h$ can be extracted as shown in Step-7 of Case $h = \mathbf{Y}$. The simulation with such D_ε and D'_ε can be done if the simulator has y-side witness and z-side witness of run_J since they differ at only one index J.

The probability distribution over these cases depends on Λ and the strategy of $\mathcal{U}_1^*$. Note that the distribution of Λ does not depend on the choice of χ as the protocol is witness indistinguishable and the public key are generated so that it distributes uniformly. Accordingly, the coin flip of χ turns the simulation to the proper case with probability 1/2. □

Lemma 3. *Any poly-time adversary $\mathcal{U}_3^*$ outputs a valid signature without interacting with $\mathcal{S}$ only with negligible probability if the discrete logarithm problem is intractable and H_1, H_2, H_3 are random oracles.*

Proof. (sketch) This is equivalent to proving the security of the ordinary (i.e., non-blind) version of the signature scheme against key-only attack [19]. Thus it can be done by the rewinding simulation in the random oracle model in a similar way as done in [28]. Given $\mathbf{Y} \in_U \langle g \rangle$, we construct a machine, $\mathcal{M}_2$, that finds $\log_g \mathbf{Y}$ in $\mathbb{Z}_q$. $\mathcal{M}_2$ first selects w, ξ randomly and sets $y = \mathbf{Y}$, $h = g^w$, $z = \mathbf{Y}g^\xi$. (Since $\mathcal{M}_2$ does not need to simulate signer $\mathcal{S}$, it can put $\mathbf{Y}$ into both y and z.) $\mathcal{M}_2$ then invokes $\mathcal{U}_3^*$ twice with the same initial settings and different ε and ε' as answers of $\mathcal{H}_3$. Let the resulting signatures be $(\zeta, \zeta_1, \rho, \varpi, \sigma_1, \sigma_2, \delta, \mu)$ and $(\zeta, \zeta_1, \rho', \varpi', \sigma'_1, \sigma'_2, \delta', \mu')$. Since $\varpi + \delta = \varepsilon \neq \varepsilon' = \varpi' + \delta'$, at least either $\varpi \neq \varpi'$ or $\delta \neq \delta'$ happens. If $\varpi \neq \varpi'$, $\mathcal{M}_2$ computes $\log_g \mathbf{Y} = \log_g y = (\rho - \rho')/(\varpi' - \varpi) \bmod q$. For the case $\delta \neq \delta'$, $\mathcal{M}_2$ computes $\gamma = \log_z \zeta = (\mu - \mu')/(\delta' - \delta) \bmod q$, $w_1 = \log_g \zeta_1 = (\sigma_1 - \sigma'_1)/(\delta' - \delta) \bmod q$, $w_2 = \log_g \zeta_2 = (\sigma_2 - \sigma'_2)/(\delta' - \delta) \bmod q$, and $\log_g \mathbf{Y} = \log_g z - \xi = (w_1 + w_2/w)/\gamma - \xi \bmod q$. □

Proof of Theorem 3. Suppose that there exists an adversary $\mathcal{U}_4^*$ that outputs $\ell + 1$ valid signatures with probability ϵ_4 not negligible in n after interacting with $\mathcal{S}$ at most ℓ times. The case of $\ell = 0$ has been proven by Lemma 3. We consider $\ell \geq 1$.

Due to Lemma 2 and 3, among the $\ell + 1$ signatures, there exist at least two signature-message pairs which contains (ζ, ζ_1) and $(\tilde{\zeta}, \tilde{\zeta_1})$ such that $\log_\zeta \zeta_1 = \log_{\tilde{\zeta}} \tilde{\zeta_1} = \log_z z_{1I}$ holds for z_{1I} used in run_I for some I in $\{1, \ldots, \ell\}$. Now, there exist two queries to $\mathcal{H}_3$ that correspond to those signatures. In a similar way as used in the proof of Lemma 2, we guess the indexes of these queries and regard $\mathcal{U}_4^*$ as being successful only if the guess is correct. Accordingly, this is equivalent to an adversary, say $\mathcal{U}_5^*$, that asks $\mathcal{H}_3$ only twice and succeeds with probability $\epsilon_5 = \epsilon_4 / \binom{q_h}{2}$ in producing two signatures in the expected relation.

We construct a machine $\mathcal{M}_3$ that, given $(\mathbf{p}, \mathbf{q}, \mathbf{g}, \mathbf{Y})$, solves $\log_g \mathbf{Y}$ in $\mathbb{Z}_q$ by using $\mathcal{U}_5^*$.

Reduction algorithm: $\mathcal{M}_3$ sets $(p, q, g) := (\mathbf{p}, \mathbf{q}, \mathbf{g})$. It then flips a coin, $\chi \in_U \{0, 1\}$, to select either $y := \mathbf{Y}$ (case $\chi = 0$) , or $y := g^x$ with randomly chosen x (case $\chi = 1$) .

1. $\mathcal{M}_3$ selects $w, w_0 \in_U \mathbb{Z}_q$ and sets $h := g^w$ and $z := g^{w_0}$ by defining $\mathcal{H}_1$ so.
2. $\mathcal{M}_3$ selects $I \in_U \{1, \ldots, \ell\}$ and $J \in_U \{1, 2\}$.
3. $\mathcal{M}_3$ runs $\mathcal{U}_5^*$ simulating $\mathcal{S}$ as follows.
   - For run_i $(i \neq I)$, $\mathcal{M}_3$ simulates with z-side witness in the same way as shown in Step-2 of Case $y = \mathbf{Y}$ in the proof of Lemma 2.
   - For run_I,
     - if $\chi = 0$, $\mathcal{M}_3$ simulates with z-side witness as above, or
     - if $\chi = 1$, it defines $z_{1I} := \mathcal{H}_2(\mathsf{rnd}_I) = \mathbf{Y}$ and follows the issuing protocol by using y-side witness.

   $\mathcal{M}_3$ simulates $\mathcal{H}_3$ by returning random values, say ε_1 and ε_2.
4. $\mathcal{U}_5^*$ outputs two signatures.
5. $\mathcal{M}_3$ rewinds and restarts $\mathcal{U}_5^*$ with the same setting. $\mathcal{M}_3$ answers J-th query to $\mathcal{H}_3$ with $\varepsilon'_J \in_U \mathbb{Z}_q$.
6. $\mathcal{U}_5^*$ outputs two signatures.
7. Let $(\zeta, \zeta_1, \rho, \varpi, \sigma_1, \sigma_2, \delta, \mu)$ and $(\zeta, \zeta_1, \rho', \varpi', \sigma'_1, \sigma'_2, \delta', \mu')$ be the resulting signatures that correspond to ε_J and ε'_J respectively. (If any of the resulting signatures does not correspond to the hash value, $\mathcal{M}_3$ fails.) If $\chi = 0$ and $\varpi \neq \varpi'$, $\mathcal{M}_3$ outputs $\log_g y = \log_g \mathbf{Y} = (\rho - \rho')/(\varpi' - \varpi) \bmod q$. If $\chi = 1$ and $\delta \neq \delta'$, it outputs $\log_g z_{1I} = \log_g \mathbf{Y} = (\sigma_1 - \sigma'_1)/(\mu - \mu') \bmod q$. $\mathcal{M}_3$ fails, otherwise.

Evaluation of success probability: (sketch)
The probability that $\mathcal{U}_5^*$ is successful and the obtained twin signatures are correlated to run_I is at least ϵ_5/ℓ. The probability is taken over the coin flips of $\mathcal{G}$, $\mathcal{S}$, $\mathcal{U}_5^*$ and the choices of $\mathcal{H}_1$, $\mathcal{H}_2$, $\mathcal{H}_3$.

According to Lemma 4, we can find, with probability at least $\epsilon_5/2\ell$, a convenient random tapes of $\mathcal{G}, \mathcal{S}, \mathcal{U}_5^*$ and $\mathcal{H}_1, \mathcal{H}_2$ that lead $\mathcal{U}_5^*$ to output twin signatures that corresponds to run_I with probability $\geq \epsilon_5/2\ell$. The success probability of $\mathcal{U}_5^*$ is now taken over the choice of $\mathcal{H}_3$, i.e., ε_1 and ε_2. We show that the standard rewinding simulation works to extract the witness of the desired side with probability not negligible in the security parameter. (The rest of the proof is actually the same as that in [3], so we give only a brief sketch below.) By $\boldsymbol{\varepsilon}$, we denote $(\varepsilon_1, \varepsilon_2)$ hereafter. Note that the number of all possible $\boldsymbol{\varepsilon}$ is q^2. Define $Succ$ as a set of $\boldsymbol{\varepsilon}$ with which $\mathcal{U}_5^*$ succeeds. Then, there exists a many-to-one mapping from $\boldsymbol{\varepsilon} \in Succ$ to e_I, which is the challenge from $\mathcal{U}_5^*$ used in run_I. Since $\epsilon_5/2\ell$ is not negligible in n, $\#Succ > q$ holds for infinitely many values of n. Thus, there exist $\boldsymbol{\varepsilon}$ and $\boldsymbol{\varepsilon}'$ in $Succ$ that result in the same e_I. Let tr_i denote a transcript obtained in run_i. That is, $tr_i = \{(\mathsf{rnd}_i, a_i, b_{1i}, b_{2i}), e_i, d_i\}$ (excluding dependent variables, r_i, w_i, s_{1i}, s_{2i}). For such $\boldsymbol{\varepsilon}$ and $\boldsymbol{\varepsilon}'$, the sequences of the transcriptions are identical with regard to run_I, that is, $(tr_1, \cdots, tr_I, \cdots, tr_\ell)$ and $(tr'_1, \cdots, tr_I, \cdots, tr'_\ell)$.

Since the issuing protocol is witness indistinguishable, the distribution of tr_I does not depend on the choice of χ. The same is true for other tr_i and tr'_i as they are produced by z-side witnesses selected independently from χ. Thus, if $\mathcal{U}_5^*$ is run twice with such $\boldsymbol{\varepsilon}$ and $\boldsymbol{\varepsilon}'$, $\mathcal{U}_5^*$ produces a collision that results in exposing either z-side witness or y-side witness independently from χ. It is successful if y-side witness is extracted when $\chi = 0$, or z-side witness, which contains $w_1 = \log_g z_1 = \log_g \mathbf{Y}$, is extracted when $\chi = 1$. These successful cases happen with probability $1/2$ due to the random choice of χ. The difficulty is that we rarely find such $\boldsymbol{\varepsilon}$ and $\boldsymbol{\varepsilon}'$. So we consider what happens if $\boldsymbol{\varepsilon}$ and $\boldsymbol{\varepsilon}''$ that result in different e_I and e'_I are chosen in the simulation. In this case, tr_I and tr'_I differ and may reflect the choice of χ so that they only yield a useless witness that we already have. We can, however, prove that such useless result cannot occur all the time. Suppose that $\chi = 0$ and $\boldsymbol{\varepsilon}$ and $\boldsymbol{\varepsilon}'$ yield y-side witness as desired, but $\boldsymbol{\varepsilon}$ and $\boldsymbol{\varepsilon}''$ only yield useless z-side witness. This means that $\varpi \neq \varpi'$ and $\varpi = \varpi''$. Thus, $\varpi' \neq \varpi''$ and desired y-side witness can be extracted if $\boldsymbol{\varepsilon}'$ and $\boldsymbol{\varepsilon}''$ are chosen. Following this observation, [3] estimated the probability of finding such a convenient pair of $\boldsymbol{\varepsilon}$ and concluded that it was not negligible in the security parameter n. □

5 Application to Double-Spender-Traceable E-cash

Here we apply the proposed blind signature scheme to create a secure anonymous e-cash scheme that provides double-spender traceability.

The withdrawal protocol is exactly the same as the signature issuing protocol. A coin is 7-tuple $\mathsf{coin} = (\zeta, \zeta_1, \rho, \varpi, \sigma_1, \sigma_2, \delta)$, which omits μ from the signature described in the previous section. The user stores the coin together with τ and γ. To pay, the user releases the coin and (ε_p, μ_p) where $\varepsilon_p = \mathcal{H}_4(z^\tau \| \mathsf{coin} \| \mathsf{desc})$ and $\mu_p = \tau - \varepsilon_p \gamma \bmod q$. Here $\mathcal{H}_4$ is a hash function $\mathcal{H}_4 : \{0,1\}^* \to \mathbb{Z}_q$ and desc is the unique description of the transaction. The shop accepts if

$$\begin{aligned} \zeta &\not\equiv 1, \\ \varpi + \delta &\equiv \mathcal{H}_3(\zeta \| \zeta_1 \| g^\rho y^\varpi \| g^{\sigma_1} {\zeta_1}^{\delta} \| h^{\sigma_2} (\zeta/\zeta_1)^{\delta} \| z^{\mu_p} \zeta^{\varepsilon_p}) \bmod q, \text{ and} \\ \varepsilon_p &\equiv \mathcal{H}_4(z^{\mu_p} \zeta^{\varepsilon_p} \| \mathsf{coin} \| \mathsf{desc}) \bmod q. \end{aligned}$$

It is not hard to see that a double payment using different desc and desc' with the same coin yields (ε_p, μ_p) and (ε'_p, μ'_p) which allows the bank to extract blinding factor γ as $\gamma = (\mu'_p - \mu_p)/(\varepsilon_p - \varepsilon'_p) \bmod p$. Since we can prove that Lemma 2 also applies to this variant, $\zeta^{1/\gamma}$ should expose z_1 used in a particular withdrawal session invoked by an authenticated user.

6 Conclusion

We presented an efficient three-move blind signature scheme. It provides one-more unforgeability with polynomially many signatures. From a practical point

of view, the scheme is less efficient than known logarithmically-secure schemes but remains practical as it costs only a few times more than the Schnorr blind signature scheme.

The unforgeability was proven under the discrete-log assumption in the random oracle model. Computing the exact reduction cost in the style of [5] seems hard due to the intricate reduction algorithm. Accordingly, the success probability was argued in a classical style, i.e., it was shown that the success probability of the reduction is not negligible with regard to the security parameter.

We also have presented a secure double-spender-traceable e-cash scheme to demonstrate the suitability of our scheme. The scheme is the first single-term scheme whose security against parallel withdrawals can be proven only under the discrete-log and the random oracle assumption.

Acknowledgments

The author wishes to thank Jan Camenisch and Eiichiro Fujisaki for their helpful comments. Early discussions with Miyako Ohkubo helped simplify the scheme.

References

1. M. Abe and J. Camenisch. Partially blind signatures. In the 1997 Symposium on Cryptography and Information Security, 1997.
2. M. Abe and E. Fujisaki. How to date blind signatures. In *Asiacrypt '96*, LNCS 1163, pp. 244–251. Springer-Verlag, 1996.
3. M. Abe and T. Okamoto. Provably secure partially blind signatures. In *Crypto 2000*, LNCS 1880, pp. 271–286. Springer-Verlag, 2000.
4. M. Bellare and P. Rogaway. Optimal asymmetric encryption. In *Eurocrypt '94*, LNCS 950, pp. 92–111. Springer-Verlag, 1995.
5. M. Bellare and P. Rogaway. The exact security of digital signatures – how to sign with RSA and Rabin. In *Eurocrypt '96*, LNCS 1070, pp. 399–416. Springer-Verlag, 1996.
6. S. Brands. Untraceable off-line cash in wallet with observers. In *Crypto '93*, LNCS 773, pp. 302–318. Springer-Verlag, 1993.
7. S. Brands. Restrictive binding of secret-key certificates. In *Eurocrypt '95*, LNCS 921, pp. 231–247. Springer-Verlag, 1995.
8. S. Brands. Restrictive blinding of secret-key certificates. Tech. report, CWI, 1995.
9. J. Camenisch. *Group Signature Schemes and Payment Systems Based on the Discrete Logarithm Problem.* PhD thesis, ETH Zürich, 1998.
10. J. Camenisch, J.-M. Piveteau, and M. Stadler. Blind signatures based on the discrete logarithm problem. In *Eurocrypt '94*, LNCS 950, pp. 428–432. Springer-Verlag, 1995.
11. J. Camenisch, J.-M. Piveteau, and M. Stadler. Fair blind signatures. In *Eurocrypt '95*, LNCS 921, pp. 209–219. Springer-Verlag, 1995.
12. D. Chaum. Blind signatures for untraceable payments. In *Crypto '82*, pp. 199–204. Prenum Publishing Corporation, 1982.
13. R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Crypto '94*, LNCS 839, pp. 174–187. Springer-Verlag, 1994.

14. I. Damgård. A design principle for hash functions. In *Crypto '89*, LNCS 435, pp. 416–427. Springer-Verlag, 1990.
15. U. Feige and A. Shamir. Witness indistinguishable and witness hiding protocols. In 21st *STOC*, pp. 416–426, 1990.
16. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Crypto '86*, LNCS 263, pp. 186–199. Springer-Verlag, 1986.
17. M. Fischlin. A note on security proofs in the generic model. In *Asiacrypt 2000*, LNCS 1976, pp. 458–469. Springer-Verlag, 2000.
18. Y. Frankel, Y. Tsiounis, and M. Yung. "Indirect discourse proofs": Achieving efficient fair off-line e-cash. In *Asiacrypt '96*, LNCS 1163, pp. 286–300. Springer-Verlag, 1996.
19. S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal of Computing*, 17(2):281–308, April 1988.
20. L. C. Guillou and J.-J. Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. In *Eurocrypt '88*, LNCS 330, pp. 123–128. Springer-Verlag, 1988.
21. H. Horster, M. Michels, and H. Petersen. Meta-message recovery and meta-blind signature schemes based on the discrete logarithm problem and their applications. In *Asiacrypt '92*, LNCS 917, pp. 224–237. Springer-Verlag, 1992.
22. A. Juels, M. Luby, and R. Ostrovsky. Security of blind digital signatures. In *Crypto '97*, LNCS 1294, pp. 150–164. Springer-Verlag, 1997.
23. T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *Crypto '92*, LNCS 740, pp. 31–53. Springer-Verlag, 1993.
24. T. Okamoto and K. Ohta. Divertible zero knowledge interactive proofs and commutative random self-reducibility. In *Eurocrypt '89*, LNCS 434, pp. 134–149. Springer-Verlag, 1990.
25. B. Pfitzmann and M. Waidner. How to break and repair a "probably secure" untraceable payment system. In *Crypto '91*, LNCS 576, pp. 338–350. Springer-Verlag, 1992.
26. D. Pointcheval. Strengthened security for blind signatures. In *Eurocrypt '98*, LNCS, pp. 391–405. Springer-Verlag, 1998.
27. D. Pointcheval and J. Stern. Provably secure blind signature schemes. In *Asiacrypt '96*, LNCS 1163, pp. 252–265. Springer-Verlag, 1996.
28. D. Pointcheval and J. Stern. Security proofs for signature schemes. In *Eurocrypt '96*, LNCS 1070, pp. 387–398. Springer-Verlag, 1996.
29. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 2000.
30. C. Schnorr and M. Jakobsson. Security of discrete log cryptosystems in the random oracle and generic model. Tech. report, University Frankfurt and Bell Labs., 1999.
31. C. P. Schnorr. Efficient signature generation for smart cards. *Journal of Cryptology*, 4(3):239–252, 1991.

Appendix

The following Lemma is known as the Heavy-row Lemma [16] or the Splitting Lemma [28,29]. Let $X \times Y$ be a product space and A its subset. Let (x, y) denote an element in $X \times Y$.

Lemma 4. *Let* A *be* $\Pr[(x,y) \in A] \geq \epsilon$ *for some* ϵ*, and* B *be* $B = \{x \in X \mid \Pr_{y \in Y}[(x,y) \in A] \geq \epsilon/2\}$*. Then,* $\Pr_{x \in X}[x \in B] \geq \epsilon/2$*.*

The following lemma is the reverse of the above in some sense.

Lemma 5. *Let* A *be* $\Pr[(x,y) \in A] < \epsilon$ *for* $\epsilon \leq 1/3$*. Define*

$$B = \{x \in X \mid \Pr_{y \in Y}[(x,y) \in A] > 1 - \epsilon\}, \text{ and}$$
$$C = \{y \in Y \mid \Pr_{x \in X}[(x,y) \in A] > 1 - \epsilon\}.$$

Then, either $\Pr[x \in B] < \epsilon$ *or* $\Pr[y \in C] < \epsilon$ *holds.*

Proof. By contradiction. Assume that $\Pr[x \in B] \geq \epsilon$ and $\Pr[y \in C] \geq \epsilon$. Let

$$BY = \{(x,y) \in A \mid x \in B\}, \text{ and}$$
$$CX = \{(x,y) \in A \mid y \in C\}.$$

Observe that $|CX| > (1-\epsilon)|X| \cdot \epsilon|Y|$ and $|BY| > \epsilon|X| \cdot (1-\epsilon)|Y|$. Let CX' and BY' denote minimal subsets of CX and BY, which, respectively, can be considered as $(1-\epsilon)|X| \times \epsilon|Y|$ and $\epsilon|X| \times (1-\epsilon)|Y|$ squares over plain $X \times Y$. Since $1 - \epsilon > \epsilon$, the maximum overlap of those squares is $\epsilon|X| \times \epsilon|Y|$. So, $|CX' \cap BY'| \leq \epsilon^2|X||Y|$. Since $|A| > |CX'| + |BY'| - |CX' \cap BY'|$, we have

$$\epsilon|X||Y| > (1-\epsilon)|X| \cdot \epsilon|Y| + \epsilon|X| \cdot (1-\epsilon)|Y| - \epsilon^2|X||Y|,$$
$$\epsilon > 1/3,$$

which is a contradiction. □

Lemma 5 can be generalized in the following way by repeatedly applying itself. Let $(x_1, \ldots, x_k)$ denote an element of product space X^k. Let $(x_1, \ldots, x_k)^j$ denote removal of the j-th element, i.e., $(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_k)^j$.

Corollary 1. *Let* A *be* $\Pr[(x_1, \ldots, x_k) \in A] < \epsilon$ *for* $\epsilon \leq 1/3$*. Then, there exists* j *such that* $\Pr[(x_1, \ldots, x_k)^j \in B_j] < \epsilon$ *where*

$$B_j = \{(x_1, \ldots, x_k)^j \mid \Pr_{x_j}[(x_1, \ldots, x_k) \in A] > 1 - \epsilon\}.$$