Provably Secure Fair Blind Signatures with Tight Revocation

Masayuki Abe[1] and Miyako Ohkubo[2]

[1] NTT Information Sharing Platform Laboratories. 1-1 Hikari-no-oka, Yokosuka-shi, 239-0847 JAPAN
abe@isl.ntt.co.jp
[2] NTT East. A-15F Shinagawa InterCity, 2-15-1 Kounan, Minato-ku, Tokyo, 108-6015 JAPAN
ookubo.miyako@east.ntt.co.jp

Abstract. A fair blind signature scheme allows the trustee to revoke blindness so that it provides authenticity and anonymity to honest users while preventing malicious users from abusing the anonymity to conduct blackmail etc. Although plausible constructions that offer efficient tricks for anonymity revocation have been published, security, especially one-more unforgeability and revocability against adaptive and parallel attacks, has not been studied well. We point out a concrete vulnerability of some of the previous schemes and present an efficient fair blind signature scheme with a security proof against most general attacks. Our scheme offers tight revocation where each signature and issuing session can be linked by the trustee.

1 Introduction

Fair blind signature schemes are a variant of blind signature schemes; they allow a trustee to revoke the blindness in such ways that

- given a view of a signature issuing session conducted with an authenticated user, the trustee can identify the resulting signature (Signature Tracing), or
- given a signature, the trustee can identify the issuing session that yielded the signature, which eventually identifies the user who conducted the session (Session Tracing).

Such schemes will play an important role in applications that must offer both privacy and authenticity while preventing users from abusing anonymity. See [25] for a concrete example.

The notion of fair blind signatures was introduced independently in [6, 9] for the construction of anonymous electronic payment schemes. Since then, some efficient constructions have been shown [23, 7] and several different approaches to the same goal have been taken [12, 16]. These previous schemes provide efficient

revocation mechanisms but their security, especially in terms of revocability and unforgeability against adaptive and parallel attacks, has not been rigorously studied. Indeed, even the security of ordinary blind signatures against parallel attacks has been studied formally only in recent works [20, 17, 22, 2, 1].

In some schemes, revocation is limited to linking a signature to its owner. There are some other schemes that allow a signature to be linked to a particular issuing session. Such a fine revocation, for instance, allows one to know the issuing time of the target signature from the session log. Typically, revocation in this type of schemes reveals the randomness generated by the user during the issuing session. Accordingly, if a malicious user broadcasts a value via the Internet and encourages all other users to use it as the random parameter in issuing sessions, revocation becomes useless. Some known schemes, e.g. [16, 7, 15], are vulnerable against this attack, or they implicitly resort to on-the-fly freshness checking, which is expensive in practice.

Our contribution is an efficient fair-blind signature scheme that is secure against adaptive and parallel attacks. Assuming the existence of ideal hash functions [5], its blindness is proven under the decision Diffie-Hellman assumption, and revocability and one-more unforgeability against adaptive and parallel attacks are proven under the discrete logarithm assumption. Another advantage of our scheme is that it offers tight revocation. That is, given a signature, revocation identifies the issuing session that uniquely produced the signature, and, given a session view, revocation identifies the unique signature created in the session. Naturally, once such tight revocability is achieved, the scheme also provides one-more unforgeability since tight and bi-directional revocability guarantees one-to-one mapping between issuing sessions and resulting signatures.

The rest of this paper is organized as follows. Section 2 defines the security of fair blind signatures. Section 3 reviews underlying ideas and building blocks. Section 4 presents our scheme in detail. A security analysis is given in Section 5. Section 6 gives several remarks. It includes weakness of our scheme, modifications, and open problems.

2 Definitions

Let $(\mathcal{G}_S, \mathcal{S}, \mathcal{U}, \mathcal{V})$ be a blind signature scheme where $\mathcal{G}_S$ is a signing key generation algorithm, $\mathcal{S}$ and $\mathcal{U}$ are interactive Turing machines called signer and user, and $\mathcal{V}$ is a signature verification algorithm. (Please refer to [17, 22] for a formal functional definition of blind signature schemes.) Informally, a fair blind signature scheme with off-line trustee is a blind signature scheme with five additional probabilistic polynomial-time algorithms, $\mathcal{G}_T$, $\mathcal{R}_{sig}$, $\mathcal{R}_{sid}$, $\mathcal{M}_{sig}$, and $\mathcal{M}_{sid}$ as follows.

$\mathcal{G}_T$ is a revocation key generation algorithm that takes a public key of a signer, say pk, and outputs a private and public revocation key pair. The keys can be independent of the public key of the signer (thus only one revocation key pair for all signers); $(rsk, rpk) \leftarrow \mathcal{G}_T(1^n, pk)$.

$\mathcal{R}_{sig}$ is a revocation algorithm that generates signature identifier I_{sig} that identifies the signature yielded from the target session. It takes the view of the signer during the target session and revocation key; $I_{sig} \leftarrow \mathcal{R}_{sig}(view_i, rsk)$.

$\mathcal{R}_{sid}$ is a revocation algorithm that generates session identifier I_{sid} that identifies the session that has produced target signature-message pair Σ_m. $I_{sid} \leftarrow \mathcal{R}_{sid}(\Sigma_m, rsk)$.

$\mathcal{M}_{sig}$ is a matching algorithm that examines whether I_{sig} matches to signature-message pair Σ_m or not. It outputs 1 if they match, 0 otherwise; $0/1 \leftarrow \mathcal{M}_{sig}(I_{sig}, \Sigma_m)$.

$\mathcal{M}_{sid}$ is a matching algorithm that examines whether I_{sid} matches to $view_i$ or not. It outputs 1 if they match, 0 otherwise; $0/1 \leftarrow \mathcal{M}_{sid}(I_{sid}, view_i)$.

These algorithms also take public data such as pk and rpk if needed. Although $view_i$ include everything that the signer can see during the session, which includes his own private key, what is really necessary to complete revocation differs $\mathcal{M}_{ss}$ differ depending on the specific revocation mechanism used.

We start the security definitions with traceability. Intuition states that a scheme is session traceable if no adversary can output a signature that can not be associated with the corresponding session, or can be associated with more than two sessions by revocation. Accordingly, it assures that each valid signature should be linked to a single session. Similarly, a scheme is signature traceable if no adversary can output two signatures that will be associated to the same session. Hence, it assures that every session should be linked to a single valid signature. If a scheme provides both types of traceability, shown below, we say that the scheme offers tight revocation.

Definition 1. *(Signature Traceability) A fair blind signature scheme is signature traceable if, for any probabilistic polynomial-time algorithm $\mathcal{U}^*$ that, after interacting with legitimate signer $\mathcal{S}$ at most ℓ times in an adaptive and arbitrarily interleaving manner, outputs*

- *a valid signature-message pair, say Σ_m, such that, for $I_{sig} = \mathcal{R}_{sig}(view_i, rsk)$, $\mathcal{M}_{sig}(I_{sig}, \Sigma_M) = 0$ holds for all $i = 1, \ldots, \ell$, or*
- *two valid and different signature-message pairs, say Σ_{m0}, Σ_{m1}, such that, there exists i in $1, \ldots, \ell$ such that $\mathcal{M}_{sig}(I_{sig}, \Sigma_{m0}) = \mathcal{M}_{sig}(I_{sig}, \Sigma_{m1}) = 1$ where $I_{sig} = \mathcal{R}_{sig}(view_i, rsk)$,*

with probability at most $1/n^c$ for sufficiently large n and some constant c. The probability is taken over the coin flips of $\mathcal{G}_S$, $\mathcal{G}_T$, $\mathcal{S}$, and $\mathcal{U}^*$.

Definition 2. *(Session Traceability) A fair blind signature scheme is session traceable if, for any probabilistic polynomial-time algorithm $\mathcal{U}^*$ that, after interacting with legitimate signer $\mathcal{S}$ at most ℓ times in an adaptive and arbitrarily interleaving manner, outputs a valid signature-message pair Σ_m such that*

- *for $I_{sid} = \mathcal{R}_{sid}(\Sigma_m, rsk)$, $\mathcal{M}_{sid}(I_{sid}, view_i) = 0$ holds for all $i = 1, \ldots, \ell$, or*
- *there exists $i, j, i \neq j$ such that $\mathcal{M}_{sid}(I_{sid}, view_i) = \mathcal{M}_{sid}(I_{sid}, view_j) = 1$,*

with probability at most $1/n^c$ *for sufficiently large* n *and some constant* c*. The probability is taken over the coin flips of* $\mathcal{G}_S$*,* $\mathcal{G}_T$*,* $\mathcal{S}$*, and* $\mathcal{U}^*$*.*

Note that, in the random oracle model, these success probabilities also depend on the choice of random oracles.

Next is blindness, which informally means that any adversary that colludes with the signer can distinguish two session views only with negligible advantage when one of the views results in a given signature.

Definition 3. *(Blindness) Let* $\mathcal{S}^*$ *and* $\mathcal{D}^*$ *be probabilistic poly-time algorithms that play the following game with honest user* $\mathcal{U}_0$ *and* $\mathcal{U}_1$*.*

1. $(pk, sk) \leftarrow \mathcal{G}_S(1^n)$*,* $(rsk, rpk) \leftarrow \mathcal{G}_T(1^n, pk)$
2. $(\mathsf{msg}_0, \mathsf{msg}_1) \leftarrow \mathcal{S}^*(sk, rpk)$
3. *For* $b \in_U \{0,1\}$*,* msg_b *is given to* $\mathcal{U}_0$*, and* msg_{1-b} *is given to* $\mathcal{U}_1$*.*
4. $\mathcal{S}^*$ *engages in the signature issuing protocol with* $\mathcal{U}_0$*,* $\mathcal{U}_1$ *in arbitrary order.*
5. *Resulting signature* Σ_0 *for* msg_0 *is given to* $\mathcal{D}^*$*.* $\mathcal{D}^*$ *also allowed to take any information from* $\mathcal{S}^*$*.*
6. $\mathcal{D}^*$ *outputs* $b' \in \{0,1\}$*.*

The 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* $\mathcal{G}_T$*,* $\mathcal{G}_S$*,* $\mathcal{S}^*$*,* $\mathcal{D}^*$ *and* $\mathcal{U}_0$*,* $\mathcal{U}_1$ *and* b*.*

Finally, we define one-more unforgeability in such a sense that it is infeasible to output $\ell + 1$ valid signatures after interacting with the signer ℓ times.

Definition 4. *(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. The interaction can be done in an adaptive and arbitrarily interleaving manner. The probability is taken over the coin flips of* $\mathcal{G}$*,* $\mathcal{S}$*, and* $\mathcal{U}^*$*.*

It is important to see that if a scheme provides tight revocability, the scheme is one-more unforgeable since tight revocability assures that there is one-to-one correspondence between successful sessions and valid signatures. Accordingly, it suffice to prove blindness and tight revocability for our scheme.

The above definitions are weak since the adversaries have no access to the trustee. Thus it is important for the trustee not to show the tracing information to anybody to prevent the adversaries from using the trustee as an oracle. When revocation is done only for private purposes such as criminal investigation, such weak definitions may suffice. Although our scheme provides security only in a weak sense, one can define a stronger notion of security by modifying the above definitions. Informally, the scheme provides *strong* signature/session traceability if traceability is retained even if the private revocation key *rsk* is given to $\mathcal{U}^*$ in Definition 1 and 2. Similarly, we say a scheme provides strong blindness if blindness is retained even if $\mathcal{S}^*$ and $\mathcal{D}^*$ are allowed to ask the trustee for revocation except for the sessions and the signature in question.

3 Underlying Idea and Building Blocks

3.1 Efficient Revocation Mechanism

We take an approach similar to that introduced in [24, 7]. Let $x_t, y_t (= g^{x_t})$ be the revocation key pair. Let z be a part of the signer's public key. To ask a signature, the user sends $(z^{1/\gamma}, g^{\gamma})$ to the signer where γ is a blinding factor that will be used later in blinding. The signer then blindly issues a signature bringing a pair $(z^{1/\gamma}, y_t)$ into the issuing protocol in such a way that a valid signature can be obtained only if the pair is blinded into $(z, {y_t}^{\gamma})$. The user can get a valid signature as he can do the conversion by taking the γ-th power. The signer is left blind since z is common to all signatures and $(y_t, g^{\gamma}, {y_t}^{\gamma})$ is assumed to be indistinguishable from $(y_t, g^{\gamma}, {y_t}^{\gamma'})$ with random γ' used for another signature.

Given a signature that contains ${y_t}^{\gamma}$, the trustee can trace the session that contains g^{γ} by computing $({y_t}^{\gamma})^{1/x_t} (= g^{\gamma})$. Similarly, given a session log that contains g^{γ}, the trustee can trace the resulting signature that must contain ${y_t}^{\gamma}$ by computing $(g^{\gamma})^{x_t} (= {y_t}^{\gamma})$.

For the above revocation mechanism to function, we must be sure that blinding by exponentiation, $(z^{1/\gamma}, y_t) \rightarrow (z, y_t^{\gamma})$, is the only way to get a valid signature. A blind signature scheme from [1] suits this purpose. As well as its security against adaptive and parallel attacks, one good property we can exploit is the restrictive blinding property. That is, when the signer issues a signature based on $(\mathsf{z}, \mathsf{z}_1)$ a user has to blind it into $(\mathsf{z}^{\gamma}, \mathsf{z}_1^{\gamma})$ to have the signature correctly blinded. So if we set $(\mathsf{z}, \mathsf{z}_1) = (z^{1/\gamma}, y_t)$, it must be transformed into $(z, {y_t}^{\gamma})$.

This trick, however, offers tight revocation only if all users are honest in choosing a unique γ in each session. Our idea for tight revocation is to add extra randomness v to the blinding factor from the signer's side so that ${y_t}^{\gamma v}$ is involved in the signature. With this adaptation, the signer can randomize blinding factor γ chosen by the user into γv so that it is unique in every session.

3.2 Verifiable Encryption of DL

For the reduction in our security proof to work, we need the trustee (simulator) to be able to extract not only ${y_t}^{\gamma}$ but also γ itself. For this purpose, a user encrypts γ with the public encryption key of the trustee and proves that γ can certainly be recovered from the ciphertext. Generally speaking, an encryption scheme accompanied by a non-interactive proof that assures the receiver that the embedded plaintext satisfies some poly-time computable predicate is often called a verifiable encryption scheme. Concrete examples can be seen in the literature, e.g. [4, 3, 8].

Let $C = (z_u, \xi) = (z^{1/\gamma}, g^{\gamma})$ be a commitment of witness γ. Let $(\mathcal{G}_E, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme. Let $(ek, dk) \leftarrow \mathcal{G}_E(1^k)$ and $E \leftarrow \mathcal{E}_{ek}(\gamma; \omega)$ where ω is a random tape. Let $\mathcal{R}$ be a relation between C and E such that

$$(C, E) \in \mathcal{R} \Leftrightarrow \log_{z_u} z \equiv \log_g \xi \equiv \mathcal{D}_{dk}(E) \bmod q.$$

Let $(\mathcal{P}, \mathcal{V})$ be a non-interactive zero-knowledge proof (argument) system for relation $\mathcal{R}$ such that $P \leftarrow \mathcal{P}(C, E, \gamma, \omega, ek)$ and $0/1 \leftarrow \mathcal{V}(C, E, P)$. We assume that it provides correctness, soundness, and computational zero-knowledge. Note that when it is zero-knowledge argument the soundness is conditionally achieved under some intractability assumptions.

On top of this standard security, we need it to be *simulatable* in such a sense that, for $C = (z^{1/\gamma}, g^{\gamma})$, there exists a poly-time simulator which, without being given γ and dk, outputs $(\tilde{E}, \tilde{P})$ such that $(C, \tilde{E}) \notin R$ and $(\tilde{E}, \tilde{P})$ is computationally indistinguishable from correct (E, P) that satisfies $(C, E) \in \mathcal{R}$ and $\mathcal{V}(C, E, P) = 1$. We say that a verifiable encryption scheme is secure and simulatable if it provides all these properties. Note that we only consider passive adversaries who have no access to the decryption oracle. When the encryption scheme is semantically secure against chosen plaintext attacks and the proof system is a public-coin honest verifier zero-knowledge proof made non-interactive with the Fiat-Shamir technique [11], simulatability is provided under the embedded assumption for the semantic security of the encryption scheme and the random oracle assumption.

Appendix A and B show two examples of verifiable encryption that provide all of the security properties we need in our construction. These schemes have different flavors. The scheme in Appendix A is taken from [3] and is based on Okamoto-Uchiyama encryption [19] combined with the statistical zero-knowledge argument of [14]. In this scheme, it is assumed that the decryption key is not given to the adversary in order to assure soundness. Accordingly, if this scheme is integrated in our construction, one has to assume that the trustee and the users are not colluding. The second scheme in Appendix B is newly constructed based on ElGamal encryption and a log-round perfect zero-knowledge proof. Though its efficiency is worse than that of the first one, this scheme provides a stronger property in that soundness holds even if the decryption key of the trustee is given to the adversary.

4 Our Scheme

[Signing Key Generation]

Let $\mathcal{G}$ be a probabilistic polynomial-time algorithm that generates a group parameter, (p, q, g, h) where p, q are primes and g, h are generators of subgroup of order q in $\mathbb{Z}_p{}^*$. A signer selects three hash functions $\mathcal{H}_1 : \{0,1\}^* \rightarrow \langle g \rangle$, $\mathcal{H}_{2,3} : \{0,1\}^* \rightarrow \{0,1\}^{|q|}$ and generates public-key $pk = (p, q, g, h, y, z)$ and private-key $sk = (x)$ as follows;

$(p, q, g, h) \leftarrow \mathcal{G}(1^n)$,
$x \in_U \mathbb{Z}_q$,
$y = g^x \bmod p$,
$z = \mathcal{H}_1(p, q, g, h, y)$.

All arithmetic operations are done in $\langle g \rangle$ hereafter unless otherwise noted.

[Revocation Key Generation]

Given the public key of a signer, the trustee generates secret-key $rsk = (x_t, dk)$ and public-key $rpk = (y_t, ek)$ where $x_t \in_U \mathbb{Z}_q^*$, $y_t = g^{x_t}$, and ek, dk are the key pair for verifiable encryption scheme described in Section 3.2.

Depending on the encryption algorithm $\mathcal{E}$ used for verifiable encryption, (ek, dk) can be common for all signers. Similarly, if (p, q, g) are common as system-wide parameters, x_t, y_t can be common, too.

[Signature Generation]

Here, we describe the signature issuing protocol in a higher level. Details can be found in Figure 1.

1. The user chooses blinding factor γ and computes $z_u = z^{1/\gamma}$ and $\xi = g^\gamma$. He then executes verifiable encryption where γ is encrypted into E and the relation among z_u, ξ, E is proven by providing P.
2. The signer verifies (E, P). He generates v randomly, and computes $z_1 = {y_t}^v$ and $z_2 = z_u/z_1$. He then proves to the user that z_1 is made as it should be by providing Schnorr zero-knowledge proof $P_s = (\sigma_s, c_s)$ where $c_s = \mathcal{H}_3(z_1 \| {y_t}^{r_s})$ and $\sigma_s = r_s - c_s v \bmod q$ for $r_s \in_U \mathbb{Z}_q$. The proof will be verified by the user as $c_s \stackrel{?}{=} \mathcal{H}_3(z_1 \| {y_t}^{\sigma_s} {z_1}^{c_s})$.
3. Based on y, z_1, z_2, the signer and the user engages in an interactive proof protocol. For the signer, the protocol is a witness indistinguishable proof of knowledge of
$$\log_g y \vee (\log_g z_1 \wedge \log_h (z_u/z_1)).$$
The signer converts the proof into the one for
$$\log_g y \vee (\log_g \zeta_1 \wedge \log_h (z/\zeta_1))$$
by exponentiating $(z_1, z_u) \stackrel{\gamma}{\to} (\zeta_1, z)$ and blinding it with the standard diversion technique [18]. The converted proof is eventually transformed to a signature with Fiat-Shamir technique.
4. The signer stores ξ^v as the identity of this session.
5. The user outputs a signature, $\Sigma = (\zeta_1, \rho, \varpi, \sigma_1, \sigma_2, \delta)$ for message m.

Note that ξ^v can be published, though it is not necessary to the user. The signer may provide extra Schnorr zero-knowledge proof that proves $\log_\xi (\xi^v) = \log_{y_t} z_1$.

[Verification]

A signature-message pair, (Σ, m), is valid if it satisfies

$$\varpi + \delta \stackrel{?}{=} \mathcal{H}_2(\zeta_1 \| g^\rho y^\varpi \| {g^{\sigma_1}} {\zeta_1}^\delta \| h^{\sigma_2} (z/\zeta_1)^\delta \| \mathsf{m}) \bmod q. \tag{1}$$

[Revocation]

Signature Tracing: Given valid (z_u, ξ, E, P) and ξ^v, the trustee computes $I_{sig} = (\xi^v)^{x_t}$. Observe that

$$I_{sig} = (\xi^v)^{x_t} = g^{\gamma v x_t} = {y_t}^{\gamma v} = \zeta_1. \tag{2}$$

Thus, I_{sig} identifies the resulting signature.

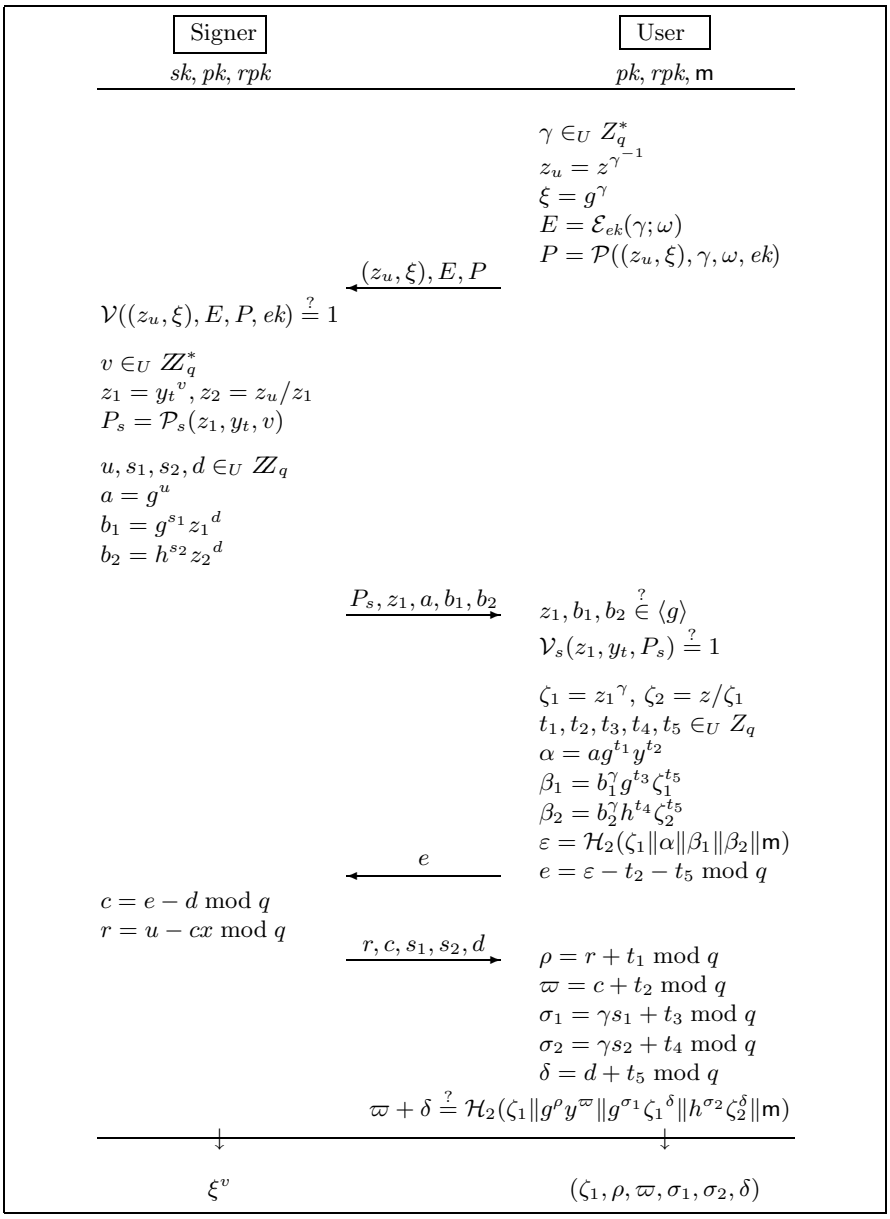

Fig. 1. The signature issuing protocol. The session aborts if any of the checks $(\stackrel{?}{=}, \stackrel{?}{\in})$ fails. $\mathcal{E}, \mathcal{P}$ are from the underlying verifiable encryption scheme of Section 3.2. $(\mathcal{P}_s, \mathcal{V}_s)$ is a Schnorr-type proof of knowledge of v w.r.t y_t and z_1. The trustee is off-line, i.e., not involved in the issuing protocol.

Session Tracing: Given a valid signature, the trustee computes $I_{ss} = \zeta_1^{1/x_t}$. Observe that

$$I_{ss} = {\zeta_1}^{1/x_t} = {z_1}^{\gamma/x_t} = {y_t}^{v\gamma/x_t} = g^{v\gamma} = \xi^v. \tag{3}$$

Since ξ^v is stored or published by the signer, I_{ss} identifies the session that issued the signature.

5 Security Proofs

5.1 Correctness

Theorem 1. *If the signer and the user follow the issuing protocol, the protocol completes with a valid signature with probability 1.*

Proof. There are four verifications denoted by $\stackrel{?}{=}$ in the issuing protocol. The verification for P and P_s in each side will accept the proof with probability 1 due to the correctness of these proof systems. It is clear that z_1, b_1, b_2 are in $\langle g \rangle$. For the last one, which is equivalent to the verification predicate, observe that the following holds.

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

Thus, the protocol always stops with a valid signature if both parties follow the protocol. □

5.2 Blindness

Theorem 2. *The proposed scheme is blind if all hash functions are random oracles, the decision Diffie-Hellman problem is intractable, and the underlying verifiable encryption scheme is secure and simulatable in the random oracle model.*

Proof. Suppose that $(\mathcal{S}^*, \mathcal{D}^*)$ is successful in breaking blindness with probability $1/2+\epsilon$ where ϵ is not negligible. We show that $\mathcal{S}^*$ and $\mathcal{D}^*$ can be used to solve the DDH problem. Define $DH = \{(X_1, X_2, X_3) \in \langle g \rangle^3 | \log_g X_1 \log_g X_2 = \log_g X_3\}$ and $RND = \{(X_1, X_2, X_3) \in \langle g \rangle^3\}$. Let $(A, B, C) \in \langle g \rangle^3$ be a DDH instance, i.e., taken from DH or RND with equal probability. Let $(A, B, C) = (g^{\mathbf{a}}, g^{\mathbf{b}}, g^{\mathbf{c}})$. If any of $\mathbf{a}, \mathbf{b}, \mathbf{c}$ is zero, we can immediately determine whether the instance is in DH or not. So we assume that none of them are zero hereafter.

Simulation proceeds as follows. We simulate hash function $\mathcal{H}_1$ so that it outputs $B^{r_{1i}}$ by selecting $r_{1i} \in_U \mathbb{Z}_q^*$ for each fresh query. Suppose that r_1 is selected for $z = \mathcal{H}_1(p, q, ...) = B^{r_1}$. Next choose $r_2 \in_u \mathbb{Z}_q^*$ and set the revocation public key as $y_t = A^{r_2}$. Select $\mathtt{d} \in_U \{0, 1\}$ and execute the issuing protocol with

$\mathcal{S}^*$ twice. Label the executions run_0 and run_1. In $\mathsf{run}_{1-\mathtt{d}}$, we simply follow the protocol. In $\mathsf{run}_{\mathtt{d}}$, we first set $z_u = g^{r_1}$ and $\xi = B$. Observe that z, z_u, and ξ are perfectly simulated no matter whether (A, B, C) is from *DH* or *RND* since z, z_u, ξ satisfies $\log_{z_u} z = \log_g \xi = (\log_g B)$. We then simulate E by encrypting $r_3 \in_U \mathbb{Z}_q^*$. Since $r_3 \neq \log_g B$ in general, $((z_u, \xi), E) \notin \mathcal{R}$. However, the simulator can produce P in such a way that (E, P) is computationally indistinguishable from the real ones since we assume that the underlying verifiable encryption is simulatable in such sense. Now send z_u, ξ, E, P and receive P_s, z_1 and etc from S^*. At this point, we rewind S^* to extract v from P_s by applying the Forking Lemma [21]. We then continue and complete the issuing session.

For message m_0 is given by $\mathcal{S}^*$ at the beginning, the simulator generates a signature-message pair, say (Σ, m_0), with regard to $\zeta_1 = C^{r_2 v}$. Other variables except for ζ_1 in Σ are generated by using the standard zero knowledge simulation technique; randomly choose $\rho, \varpi, \sigma_1, \sigma_2, \delta$, and then freely define $\mathcal{H}_2$ so that they look consistent. Given (Σ, m) and views from $\mathcal{S}^*$, distinguisher $\mathcal{D}^*$ outputs $\mathtt{d}'$. If $\mathtt{d}' = \mathtt{d}$, we conclude that (A, B, C) is in *DH*. It is in *RND*, otherwise.

We now claim that if $(A, B, C) \in$ *DH*, Σ is a valid signature that could have been produced in $\mathsf{run}_{\mathtt{d}}$. Observe that, for z, z_1 used in run_b,

$$(z_u, z, z_1, \zeta_1) = (g^{r_1}, g^{\mathbf{b} r_1}, g^{\mathbf{a} r_2 v}, g^{\mathbf{c} r_2 v}).$$

So if $\mathbf{ab} = \mathbf{c}$, we have a consistent blinding factor, $\gamma = \mathbf{b}$ which satisfies $\gamma = \log_{z_u} z = \log_{z_1} \zeta_1$ for z_u and z_1 used in $\mathsf{run}_{\mathtt{d}}$. Furthermore, there are blinding factors t_1, t_2, t_3, t_4, t_5 that convert the view of $\mathsf{run}_{\mathtt{d}}$ into the remaining elements in Σ. On the other hand, Σ could have been produced by $\mathsf{run}_{1-\mathtt{d}}$ only with negligible probability as z_u, z_1 should differ in $\mathsf{run}_{1-\mathtt{d}}$. Accordingly, given Σ, $\mathcal{D}^*$ outputs $\mathtt{d}' = \mathtt{d}$ with probability $1/2 + \epsilon$.

Next, we claim that if $(A, B, C) \in$ *RND*, Σ is statistically independent of the views of the signer in run_0 and run_1 since $\log_{z_u} z \neq \log_{z_1} \zeta_1$ holds for (z_u, z_1) in both runs except with negligible probability. Hence, $\mathtt{d}$ is also statistically independent of the view of the signer, and $\mathtt{d}' = \mathtt{d}$ happens with probability close to $1/2$ except for a negligible fraction.

In total, the success probability is $1/2(1/2+\epsilon)+1/2(1/2) = 1/2+\epsilon/2$, which contradicts the DDH assumption when ϵ is not negligible.

□

5.3 Tight Revocability

Theorem 3. *The proposed scheme is session traceable if all hash functions are random oracles, the discrete logarithm is intractable, and the soundness condition of the underlying verifiable encryption scheme holds.*

Proof. Here we must show two properties. We first show that it is infeasible for a user to produce a signature $\Sigma^\star = (\zeta_1, \rho, \varpi, \sigma_1, \sigma_2, \delta)$ such that $\log_{z_u} z \neq \log_{z_1} \zeta_1$ for all (z_u, z_1) used in issuing sessions. We then show that a valid signature cannot be linked to more than one session.

Assume that having at most q_h accesses to $\mathcal{H}_2$ and asking at most ℓ signatures to $\mathcal{S}$, $\mathcal{U}_0^*$ outputs signature $\Sigma^\star = (\zeta_1, \rho, \varpi, \sigma_1, \sigma_2, \delta)$ that satisfies $\log_{z_u} z \neq \log_{z_1} \zeta_1$ for (z_u, z_1) used in any session. Here, q_h and ℓ are bound by a polynomial of security parameter n. Let ϵ_0 be the success probability of $\mathcal{U}_0^*$, which is not negligible in 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 $\mathcal{H}_2$. (If it does not correspond to any query, $\mathcal{U}_0^*$ is successful only with negligible probability due to the randomness of $\mathcal{H}_2$.) Accordingly, it is equivalent to assuming an adversary, say $\mathcal{U}_1^*$, that asks $\mathcal{H}_2$ only once and succeeds with probability $\epsilon_1 \geq \epsilon_0/q_h$. By using $\mathcal{U}_1^*$, we construct machine $\mathcal{M}_1$ that solves the discrete-log problem. Let $(\mathbf{p}, \mathbf{q}, \mathbf{g}, \mathbf{Y})$ be an instance of the discrete-log problem to solve $\mathbf{X} = \log_{\mathbf{g}} \mathbf{Y}$ in $\mathbb{Z}_{\mathbf{q}}$.

Reduction Algorithm: $\mathcal{M}_1$ first sets $(p, q, g) := (\mathbf{p}, \mathbf{q}, \mathbf{g})$. It also generates key pair (dk, ek) for the underlying verifiable encryption scheme. 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 $\chi = 0$:

Intuition: We set $y = \mathbf{Y}$ and attempt to extract the y-side witness by simulating the signing oracle with z-side witness, which is $\log_g z_1$ and $\log_h z_2$. We run $\mathcal{U}_1^*$ twice with a different answer from $\mathcal{H}_2$ and apply the Forking Lemma. It should cause a change of either δ or ϖ in the resulting signatures. If we are lucky, we have different ϖ's and can extract the y-side witness.

1. $\mathcal{M}_1$ sets $y = \mathbf{Y}$.
2. $\mathcal{M}_1$ selects $w, w_0, w_1 \in_U \mathbb{Z}_q^*$ and sets $h := g^w$, $z := \mathcal{H}_1(p\|q\|g\|y) = g^{w_0}$, and $y_t = g^{w_1}$.
3. $\mathcal{M}_1$ runs $\mathcal{U}_1^*$ and simulates $\mathcal{S}$ for i-th query in the following way.
   (a) Given $(z_{ui}, \xi_i, E_i, P_i)$ from the user, check P_i and reject if incorrect. Otherwise, decrypt $E_i \to \gamma_i$.
   (b) Compute $a_i := g^{r_i} y^{c_i}$ for $c_i, r_i \in_U \mathbb{Z}_q$.
   (c) Compute $w_{1i} = w_1 v_i \bmod q$ and $w_{2i} = (w_0/\gamma_i - w_{1i})/w \bmod q$ for $v_i \in_U \mathbb{Z}_q^*$. Then set $z_{1i} = g^{w_{1i}}$ and $z_{2i} = h^{w_{2i}}$.
   (d) Compute P_{si} by using legitimate witness v_i.
   (e) Compute $b_{1i} := g^{u_{1i}}$ and $b_{2i} := h^{u_{2i}}$ with $u_{1i}, u_{2i} \in_U \mathbb{Z}_q$.
   (f) Send $P_{si}, a_i, b_{1i}, b_{2i}$ to $\mathcal{U}_1^*$.
   (g) 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 $s_{2i} := u_{2i} - d_i w_{2i} \bmod q$.
   (h) Send $r_i, c_i, s_{1i}, s_{2i}, d_i$ to $\mathcal{U}_1^*$.

   $\mathcal{M}_1$ simulates $\mathcal{H}_2$ by returning $\varepsilon \in_U \mathbb{Z}_q$.
4. $\mathcal{U}_1^*$ outputs a signature, say $(\zeta_1, \rho, \varpi, \sigma_1, \sigma_2, \delta)$, that corresponds to ε.
5. Reset and restart $\mathcal{U}_1^*$ with the same setting. $\mathcal{M}_1$ simulates $\mathcal{H}_2$ with $\varepsilon' \in_U \mathbb{Z}_q$. In this second run, $\mathcal{M}_1$ also uses the same random tape.
6. $\mathcal{U}_1^*$ outputs a signature, say $(\zeta_1, \rho', \varpi', \sigma_1', \sigma_2', \delta')$, that corresponds to ε'.
7. If $\varpi \neq \varpi'$, $\mathcal{M}_1$ outputs $X := (\rho - \rho')/(\varpi' - \varpi) \bmod q$. The simulation fails, otherwise.

Case $\chi = 1$:

Intuition: We set $h = \mathbf{Y}$, $z = g^{w_1} h^{w_2}$ with random w_1, w_2, and attempt to extract different representation of z, that leads $\log_g h$. The signing oracle is simulated with y-side witness except for one query. For the one randomly chosen J-th query, we use y-side witness and z-side witness, i.e., (w_1, w_2), together. We rewind $\mathcal{U}_1^*$ to apply the Forking Lemma. But this time, we fork the process by changing d in the J-th issuing session, which is used as a challenge to the z-side proof. We can answer to two different d's in the J-th session since the z-side witness in this session is (w_1, w_2). Now if δ is sensitive to the change of d, we have different δ's and can extract the z-side witness which is different from (w_1, w_2).

1. $\mathcal{M}_1$ sets $h = \mathbf{Y}$.
2. $\mathcal{M}_1$ 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 $z := \mathcal{H}_1(p\|q\|g\|y) = g^{w_1} h^{w_2}$.
3. $\mathcal{M}_1$ selects $J \in_U \{1, \ldots, \ell\}$. It also selects v_J and set $y_t = g^{w_1/v_J}$.
4. $\mathcal{M}_1$ runs $\mathcal{U}_1^*$ and simulates the signing oracle for the i-th query in the following way.
   (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 x and (w_1, w_2) as follows.
      i. Given $(z_{ui}, \xi_i, E_i, P_i)$ from the user, check P_i and reject if incorrect. Otherwise, decrypt $E_i \to \gamma_i$.
      ii. Set $z_{1J} = {y_t}^{v_J}$. (Accordingly, $z_{1J} = g^{w_1}$ and $z_{2J} = h^{w_2}$.)
      iii. Compute $a_J = g^{u_J}, b_{1J} = g^{u_{1J}}, b_{2J} = h^{u_{2J}}$ with $u_J, u_{1J}, u_{2J} \in_U \mathbb{Z}_q$.
      iv. Send $(v_J, a_J, b_{1J}, b_{2J})$ to $\mathcal{U}_1^*$.
      v. 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 $s_{2J} := u_{2J} - d_J w_2 \bmod q$.
      vi. Send $(r_J, c_J, s_{1J}, s_{2J}, d_J)$ to $\mathcal{U}_1^*$.

   $\mathcal{M}_1$ simulates $\mathcal{H}_2$ by returning $\varepsilon \in_U \mathbb{Z}_q$.
5. $\mathcal{U}_1^*$ outputs a signature, say $(\zeta_1, \rho, \varpi, \sigma_1, \sigma_2, \delta)$, that corresponds to ε.
6. Rewind and restart $\mathcal{U}_1^*$ with the same setting. Then choose $I \in_U \{0, \ldots, \ell\}$.
   – If $I = 0$, $\mathcal{M}_1$ simulates $\mathcal{H}_2$ by returning $\varepsilon' \in_U \mathbb{Z}_q$. Otherwise, set $\varepsilon' = \varepsilon$.
   – If $I \neq 0$ and run_J have not yet been completed before the query to $\mathcal{H}_2$ 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 $d'_J = d_J$.
7. $\mathcal{U}_1^*$ outputs a signature, say $(\zeta_1, \rho', \varpi', \sigma'_1, \sigma'_2, \delta')$, that corresponds to ε'.
8. If $\delta = \delta'$, simulation fails. Otherwise, $\mathcal{M}_1$ computes $w'_1 = (\sigma_1 - \sigma'_1)/(\delta' - \delta) \bmod q$, $w'_2 = (\sigma_2 - \sigma'_2)/(\delta - \delta') \bmod q$, and outputs $\mathbf{X} = (w_1 - w'_1)/(w'_2 - w_2) \bmod q$.

Sketch of success probability evaluation:
Suppose that all random variables chosen by the simulating signer are determined

purely from the random tape so that they are fixed before the simulation starts. We consider how δ in $\Sigma^\star$ is sensitive to the alteration of ε and $\{d_{i_{k+1}}, \ldots, d_{i_\ell}\}$ which are given after ε is given to $\mathcal{U}_1^*$. Observe that independent variables given to $\mathcal{U}_1^*$ are p, q, g, h, y, $\mathcal{H}_1$, $\mathcal{H}_2$, $\mathsf{sid}_i, a_i, b_{1i}, b_{2i}$, 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 and outputs of $\mathcal{U}_1^*$. We wrap all these independent variables into Λ, except for $\{\varepsilon, d_{i_{k+1}}, \ldots, d_{i_\ell}\}$, which is denoted by D_ε hereafter. 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 the Splitting Lemma [11, 22], 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 $(\Lambda, D_\varepsilon) \in S$ to δ. If $(\Lambda, D_\varepsilon) \notin S$, we denote $\perp \leftarrow D_\varepsilon$.

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 $\Sigma^\star$. Let $\psi_{max} = \psi(\delta_{max})$. We consider two cases.

Case 1 (ψ_{max} is not negligible) :

In this case, for randomly chosen D_ε and D'_ε, the adversary is likely to output signatures that contain δ_{max} with sufficiently large probability. When δ is the same for different ε from $\mathcal{H}_2$, ϖ must differ as $\delta + \varpi = \varepsilon$. Consequently, with sufficient probability, we obtain $\varpi \neq \varpi'$ with which y-side witness can be extracted as written in Step-7 of Case $\chi = 0$. For more details, we refer to the proof of Lemma 3 of [1].

Case 2 (ψ_{max} is negligible) :

In this case, δ tends to change if D_ε is altered. Due to [1], randomly chosen D_ε and D'_ε that differ only at one position lead $\mathcal{U}_1^*$ to output two corresponding signatures $(\zeta_1, \rho, \varpi, \sigma_1, \sigma_2, \delta)$ and $(\zeta_1, \rho', \varpi', \sigma'_1, \sigma'_2, \delta')$ with sufficiently large probability. From these signatures, we can extract w'_1, w'_2 that satisfy $\zeta_1 = g^{w'_1}$ and $\zeta/\zeta_1 = h^{w'_2}$. By assumption, $\log_{z_u} z \neq \log_{z_1} \zeta_1$. So $w_1 \neq w'_1$ and $w_2 \neq w'_2$ holds. Accordingly $\mathbf{X} = \log_g h = (w_1 - w'_1)/(w'_2 - w_2) \bmod q$ is computable.

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 is generated so that it distributes uniformly. Accordingly, the coin flip of χ turns the simulation to the proper case with probability 1/2.

In the above, we proved that for ζ_1 in a valid signature, there exists at least one session that includes (z_u, z_1) that satisfies $\log_{z_u} z = \log_{z_1} \zeta_1$. Since $z_1 (= {z_u}^v)$ depends on random v chosen by the honest signer and z_u is in $\langle g \rangle$ when P is valid, z_1 is unique among all sessions with overwhelming probability if only polynomially many sessions are executed.

We also need to prove that a signature cannot be produced without interacting with the legitimate signer. This can be done by a standard argument that uses the Forking Lemma and so is omitted here.

Finally, we need to show that a session that includes target (z_u, z_1) can be identified from ζ_1^{1/x_t}. For this, observe that the rightmost equality in Equation 3 holds because $\xi = g^\gamma$ for $\gamma = \log_{z_u} z = \log_{z_1} \zeta_1$ with overwhelming probability due to the soundness of P. □

Theorem 4. *The proposed scheme is signature traceable if all hash functions are random oracles, the discrete logarithm is intractable, and the soundness condition of the verifiable encryption scheme holds.*

Proof. We need to show that no adversary can generate a signature containing ζ_1 such that $\zeta_1 \neq (\xi^v)^{x_t}$ for any (ξ^v) stored by the signer. This can be done in the same way as done in the proof of Theorem 3.

In the following, we show that it is infeasible for the user to output two valid signatures that contain the same ζ_1 regardless of the user's behaviour.

The proof is done by contradiction. Suppose that there exists an adversary $\mathcal{U}_2^*$ that outputs two valid signatures that result in the same session by revocation with success probability ϵ_2. Here, ϵ_2 is not negligible in n and $\mathcal{U}_2^*$ is allowed to interact with $\mathcal{S}$ at most ℓ times in an arbitrary fashion. Let $\ell \geq 1$. ($\ell = 0$ was considered in Theorem 3.)

Now there exist two queries to $\mathcal{H}_2$ that correspond to those two signatures. In a similar way as used in the proof of Theorem 3, we guess the indexes of these queries and regard $\mathcal{U}_2^*$ as being successful only if the guess is correct. Accordingly, this is equivalent to an adversary, say $\mathcal{U}_3^*$, that asks $\mathcal{H}_2$ only twice and succeeds with probability $\epsilon_3 = \epsilon_2 / \binom{q_h}{2}$ in producing two signatures in the expected relation.

We construct a machine $\mathcal{M}_2$ that, given $(\mathbf{p}, \mathbf{q}, \mathbf{g}, \mathbf{Y})$, solves $\mathbf{X} = \log_g \mathbf{Y}$ in $\mathbb{Z}_q$ by using $\mathcal{U}_3^*$.

Reduction algorithm:

1. $\mathcal{M}_2$ sets $(p, q, g) := (\mathbf{p}, \mathbf{q}, \mathbf{g})$.
2. $\mathcal{M}_2$ sets either $y = \mathbf{Y}$ or $y = g^x$ for $x \in_U \mathbb{Z}_q^*$ by flipping coin χ.
3. $\mathcal{M}_2$ selects $w, w_0, w_1 \in_U \mathbb{Z}_q$ and sets $h := g^w$ and $z := g^{w_0}$, $y_t := g^{w_1}$.
4. $\mathcal{M}_2$ selects $I \in_U \{1, \ldots, \ell\}$.
5. $\mathcal{M}_2$ runs $\mathcal{U}_3^*$ simulating $\mathcal{S}$ as follows.
   - For run_i $(i \neq I)$, $\mathcal{M}_2$ simulates with z-side witness in the same way as shown in Step-3 of Case $\chi = 0$ in the proof of Theorem 3.
   - For run_I,
     - if $y = \mathbf{Y}$, $\mathcal{M}_2$ simulates with z-side witness as above, otherwise
     - it sets $z_{1I} = \mathbf{Y}$ and simulate P_s in the standard way by setting $\mathcal{H}_3$ conveniently. Then follow the rest of the protocol using x. Save γ_I by decrypting E_I.

   $\mathcal{M}_2$ simulates $\mathcal{H}_2$ by returning random values, say ε_1 and ε_2.
6. $\mathcal{U}_3^*$ outputs two signatures.

7. $\mathcal{M}_2$ rewinds and restarts $\mathcal{U}_3^*$ with the same setting. It selects $J \in_U \{1, 2\}$ and answers to J-th query to $\mathcal{H}_2$ with $\varepsilon'_J \in_U \mathbb{Z}_q$.
8. $\mathcal{U}_3^*$ outputs two signatures.
9. Let $(\zeta_1, \rho, \varpi, \sigma_1, \sigma_2, \delta)$ and $(\zeta_1, \rho', \varpi', \sigma'_1, \sigma'_2, \delta')$ 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, $\mathcal{M}_2$ fails.) If $\chi = 0$ and $\varpi \neq \varpi'$, $\mathcal{M}_2$ outputs $\log_g y = \log_g \mathbf{Y} = (\rho - \rho')/(\varpi' - \varpi) \bmod q$. If $\chi = 1$ and $\delta \neq \delta'$, it outputs $\log_g z_{1I} = \log_g \mathbf{Y} = (\sigma_1 - \sigma'_1)/\gamma_I(\delta - \delta') \bmod q$. $\mathcal{M}_2$ fails, otherwise.

We omit the evaluation of success probability as it can be done in the same way as shown in the proof of Theorem 3 of [1]. □

Due to Theorem 4 and 3, the mapping between each session and valid signature is bijective with overwhelming probability. Accordingly, we have the following corollary.

Corollary 1. *The proposed scheme is $(\ell, \ell + 1)$-unforgeable for polynomially bound ℓ if the discrete logarithm is intractable, all hash functions are random oracles, and the verifiable encryption is secure and simulatable.*

6 Remarks and Open Problems

- When each user uses a unique (z_u, ξ, E, P) repeatedly in all issuing sessions, i.e. as a public-key of the user, the scheme provides blindness (and unlinkability) in a weak sense. That is, signatures are computationally independent of each other unless the signer cooperates with the attacker. Such low-level privacy may be acceptable in applications as it offers less computation and communication complexity instead.
- As briefly mentioned in Section 2, the security definitions and the proofs confirm the security under the assumption that the trustee will never be abused as an oracle. Accordingly, the trustee must not show the tracing information to anybody. To provide stronger security in blindness where the trustee can publish the tracing information, we need the following properties. First, the verifiable encryption must be non-malleable against adaptive chosen message attacks. It also has to provide public verifiability. Second, the signature scheme must be unforgeable even for the signer in such a sense that for target signature Σ produced from a session identified by ξ^v the signer should not be able to produce valid signature $\Sigma'(\neq \Sigma)$ that results in tracing information that is relative to ξ^v. This property is not achieved in our construction even if we restrict Σ' to be different from Σ in the part necessary for revocation, which is ζ_1 in our case. A particular attack on the strong blindness is as follows. The signer transforms ζ_1 in challenge signature Σ into $\zeta' = \zeta_1^a$ with random a and creates signature Σ' that includes ζ' by using real signing key x. Session tracing information computed from ζ' will be $(\xi^v)^a$ and the signer can obtain target session identifier ξ^v. This particular attack can be prevented but we leave a provably secure solution for this issue an open problem.

- It is important to point out that, since the trustee can recover γ from E, he can produce signature Σ' that results in the same tracing information ξ^v linked from signature Σ legitimately produced by the user. Such a threat can be eliminated by encrypting γ with a encryption key whose decryption key is not known to anybody. (Remember that the decryption-key is not necessary for the trustee to complete revocation.) But for the sake of security proof, the simulator must be able to decrypt it. This is possible, for instance, with the verifiable encryption scheme in Appendix B. By generating encryption key y as $y = \mathcal{H}(\mathsf{str})$ where str is a fixed public string and $\mathcal{H}$ is a hash function $\mathcal{H} : \{0,1\}^* \rightarrow \langle g \rangle$. In this way, any party can be convinced that no one knows the decryption key corresponding to y, but a simulator that simulates the hash function as a random oracle in the proof of revocability can assign arbitrary g^x as $\mathcal{H}(\mathsf{str})$ so that x is known only to the simulator.
- Since revocation only identifies a specific randomness appearing in a issuing session, it would be necessary to assure that the session is really done by the user. An easy solution would be to have the transcript signed by the user. Although the signer may flame the user by creating Σ' from Σ so that they result in the same session tracing information in the similar way shown in the second remark, one can see that it is not the user who created the second signature due to Theorem 3.

Acknowledgements

The authors thank David Pointcheval for helpful comments. Contribution from Eiichiro Fujisaki about verifiable encryption schemes is appreciated very much.

References

1. M. Abe. A three-move blind signature scheme secure for pollynomially many signatures. In B. Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT '01*, volume 2045 of *Lecture Notes in Computer Science*, pages 136–151. Springer-Verlag, 2001.
2. M. Abe and T. Okamoto. Provably secure partially blind signatures. In M. Bellare, editor, *Advances in Cryptology — CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 271–286. Springer-Verlag, 2000.
3. G. Ateniese. Efficient verifiable encryption (and fair exchange) of digital signatures. In *ACM CCS'99*, pages 138–146. Association for Computing Machinery, 1999.
4. F. Bao. An efficient verifiable encryption scheme for encryption of discrete logarithms. In *CARDIS'98*, 1998.
5. M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *First ACM Conference on Computer and Communication Security*, pages 62–73. Association for Computing Machinery, 1993.
6. E. Brickell, P. Gemmell, and D. Kravitz. Trustee-based tracking extensions to anonymous cash and the making of anonymous change. In *Proceedings of Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 457–466. ACM, 1995.

7. J. Camenisch. *Group Signature Schemes and Payment Systems Based on the Discrete Logarithm Problem*. PhD thesis, ETH Zürich, 1998.
8. J. Camenisch and I. Damgård. Verifiable encryption, group encryption, and their applications to separable group signatures and signature sharing schemes. In T. Okamoto, editor, *Advances in Cryptology – Asiacrypt 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 331–345. Springer-Verlag, 2000.
9. J. Camenisch, J.-M. Piveteau, and M. Stadler. Fair blind signatures. In L. C. Guillou and J.-J. Quisquater, editors, *Advances in Cryptology — EUROCRYPT '95*, volume 921 of *Lecture Notes in Computer Science*, pages 209–219. Springer-Verlag, 1995.
10. D. L. Chaum and T. P. Pedersen. Wallet databases with observers. In E. F. Brickell, editor, *Advances in Cryptology — CRYPTO '92*, volume 740 of *Lecture Notes in Computer Science*, pages 89–105. Springer-Verlag, 1993.
11. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *Advances in Cryptology — CRYPTO '86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–199. Springer-Verlag, 1987.
12. Y. Frankel, Y. Tsiounis, and M. Yung. "Indirect discourse proofs": Achieving efficient fair off-line e-cash. In K. Kim and T. Matsumoto, editors, *Advances in Cryptology — ASIACRYPT '96*, volume 1163 of *Lecture Notes in Computer Science*, pages 286–300. Springer-Verlag, 1996.
13. E. Fujisaki. A simple approach to secretly sharing a factoring witness in publicly-verifiable manner. (unpublished manuscript), 2001.
14. E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In B. S. Kaliski Jr., editor, *Advances in Cryptology — CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 16–30. Springer-Verlag, 1997.
15. M. Jakobsson and J. Müller. Improved magic ink signatures using hints. In *Financial Cryptography'99*, 1999.
16. M. Jakobsson and M. Yung. Distributed "Magic Ink" signatures. In W. Fumy, editor, *Advances in Cryptology — EUROCRYPT '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 450–464. Springer-Verlag, 1997.
17. A. Juels, M. Luby, and R. Ostrovsky. Security of blind digital signatures. In B. S. Kaliski Jr., editor, *Advances in Cryptology — CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 150–164. Springer-Verlag, 1997.
18. T. Okamoto and K. Ohta. Divertible zero knowledge interactive proofs and commutative random self-reducibility. In J.-J. Quisquater and J. Vandewalle, editors, *Advances in Cryptology – EUROCRYPT '89*, volume 434 of *Lecture Notes in Computer Science*, pages 134–149. Springer-Verlag, 1990.
19. T. Okamoto and S. Uchiyama. A new public-key cryptosystem as secure as factoring. In K. Nyberg, editor, *Advances in Cryptology — EUROCRYPT '98*, volume 1403 of *Lecture Notes in Computer Science*, pages 308–318. Springer-Verlag, 1998.
20. D. Pointcheval and J. Stern. Provably secure blind signature schemes. In K. Kim and T. Matsumoto, editors, *Advances in Cryptology – ASIACRYPT '96*, volume 1163 of *Lecture Notes in Computer Science*, pages 252–265. Springer-Verlag, 1996.
21. D. Pointcheval and J. Stern. Security proofs for signature schemes. In U. Maurer, editor, *Advances in Cryptology — EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 387–398. Springer-Verlag, 1996.
22. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 2000.

23. M. Stadler. *Cryptographic Protocols for Revocable Privacy.* PhD thesis, Swiss Federal Institute of Technology Zürich, 1996.
24. M. Stadler. Publicly verifiable secret sharing. In U. Maurer, editor, *Advances in Cryptology — EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 190–199. Springer-Verlag, 1996.
25. S. von Solms and D. Naccache. On blind signatures and perfect crime. *Computer & Security*, 11:581–583, 1992.
26. A. Young and M. Yung. Finding length-3 positive cunningham chains and their cryptographic significance. In *ANTS '98*, Lecture Notes in Computer Science. Springer-Verlag, 1998.

Appendix A

The following verifiable encryption scheme is taken from [13]. Let $(\mathsf{n}, \mathsf{g}, \mathsf{h}, \ell_{\mathsf{g}})$ be the public key and (p, q) be the secret key of the Okamoto-Uchiyama encryption scheme. Here, $\mathsf{n} = \mathsf{p}^2\mathsf{q}$, and g is in $\mathbb{Z}_{\mathsf{n}}$ that satisfies $\mathrm{ord}(\mathsf{g} \bmod \mathsf{p}^2) = \mathsf{p}(\mathsf{p}-1)$, $\mathsf{h} = {\mathsf{h}_0}^{\mathsf{n}} \bmod \mathsf{n}$ for randomly chosen $\mathsf{h}_0 \in \mathbb{Z}_n$, and ℓ_{g} is the bit length of the order of g. We assume that $\ell_g > 2\ell_q$ where ℓ_q is the bit length of q. Let $\mathcal{H}_4 : \{0,1\}^* \rightarrow \{0,1\}^{\ell_q}$ be a hash function.

Now, γ is encrypted by Okamoto-Uchiyama encryption as $E = \mathsf{g}^{\gamma}\mathsf{h}^{t_u} \bmod \mathsf{n}$ where $t_u \in_U \mathbb{Z}_n$. For $C = (z_u, \xi) = (z^{1/\gamma}, g^{\gamma})$, $(C, E) \in \mathcal{R}$ is proven by providing $P = (c_u, s_{1u}, s_{2u})$ computed by the prover as follows.

1. Choose $k_1 \in_U \{0,1\}^{\epsilon_s \ell_g}$ and $k_2 \in_U \{0,1\}^{\epsilon_s(\ell_g+\ell_q)}$.
2. Compute $c_u = \mathcal{H}_4(z_u, \xi, E, {z_u}^{k_1} \bmod p, {y_t}^{k_1} \bmod p, \mathsf{g}^{k_1}\mathsf{h}^{k_2} \bmod \mathsf{n})$.
3. Compute $s_1 = k_1 - c_u\gamma$ and $s_2 = k_2 - c_u t_u$ in $\mathbb{Z}$.

Here ϵ_s is a security parameter larger than 1. P is valid if it satisfies

$$
\begin{aligned}
&c_u \in \{0,1\}^{\ell_q}, \\
&s_{1u} \in \{0,1\}^{\epsilon_s \ell_{\mathsf{g}}}, \text{ and} \\
&c_u = \mathcal{H}_4(z_u, \xi, E, {z_u}^{s_{1u}} g^{c_u} \bmod p, {y_{t_u}}^{s_{2u}} \xi^{c_u} \bmod p, \mathsf{g}^{s_{1u}}\mathsf{h}^{s_{2u}}E^{c_u} \bmod \mathsf{n}).
\end{aligned}
$$

The above protocol is a statistical zero-knowledge argument for relation R. Soundness is due to the strong RSA assumption over n. The detailed security proof can be found in [13].

Appendix B

In this section, we require that $p = 2q+1$ and $q = 2s+1$ for prime s. (See [26] for generating such Cunningham Chains.) Let h be a generator of a prime subgroup in $\mathbb{Z}_q$ where $\mathrm{ord}(\mathsf{h}) = s$. Let $(\mathsf{x}, \mathsf{y}) \in \mathbb{Z}_s \times \langle \mathsf{h} \rangle$ be a key pair of ElGamal encryption defined over $\langle \mathsf{h} \rangle$. That is, $\mathsf{y} = \mathsf{h}^{\mathsf{x}} \bmod q$.

For $\gamma \in \mathbb{Z}_q$ and $C = (z_u, \xi) = (z^{1/\gamma}, g^{\gamma})$, (E, P) is computed as follows. We first transform γ into $\gamma^{\star} \in \langle \mathsf{h} \rangle$ by

$$\gamma^{\star} = J_q(\gamma) \cdot \gamma \bmod q.$$

Here, $J_q(\gamma)$ is the Jacobi symbol, $(\frac{\gamma}{q})$. $\gamma^\star$ is then encrypted into $E = (C_1, C_2)$ using ElGamal encryption as

$$C_1 = \gamma^\star \cdot \mathsf{y}^\omega \bmod q,$$
$$C_2 = \mathsf{h}^\omega \bmod q,$$

where $\omega \in_U \mathbb{Z}_s$. When E is decrypted into $\gamma^\star$ and $g^{\gamma^\star} \bmod p \neq \xi$, γ is obtained by $\gamma = -1 \cdot \gamma^\star \bmod q$. Otherwise, $\gamma = \gamma^\star$.

The proof is done in two steps. In the first step, the prover proves relation $\log_{z_u} z = \log_g \xi$ by the Chaum-Pedersen protocol [10]. In the second step, we prove in zero-knowledge manner that $\mathcal{D}(E) = J_q(\log_g \xi) \cdot \log_g \xi \bmod q$ by repeating the following protocol sufficiently many times.

1. The prover selects $a \in_U \mathbb{Z}_q^*$ and $b \in_U \mathbb{Z}_s$ and sends

$$T_0 = \xi^a \bmod p,$$
$$T_1 = C_1 \cdot J_q(a) \cdot a \cdot \mathsf{y}^b \bmod q, \text{ and}$$
$$T_2 = C_2 \cdot \mathsf{h}^b \bmod q$$

   to the verifier.
2. The verifier sends $c \in_U \{0, 1\}$ to the prover.
3. The prover sends (α, β) where $(\alpha, \beta) = (a, b)$ when $c = 0$, and $(\alpha, \beta) = (a\gamma \bmod q, b + \omega \bmod q)$ when $c = 1$.
4. The verifier accepts if, for $c = 0$,

$$T_0 = \xi^\alpha \bmod p,$$
$$T_1 = C_1 \cdot J_q(\alpha) \cdot \alpha \cdot \mathsf{y}^\beta \bmod q, \text{ and}$$
$$T_2 = C_2 \cdot \mathsf{h}^\beta \bmod q,$$

   and for $c = 1$,

$$T_0 = g^\alpha \bmod p,$$
$$T_1 = J_q(\alpha) \cdot \alpha \cdot \mathsf{y}^\beta \bmod q, \text{ and}$$
$$T_2 = \mathsf{h}^\beta \bmod q.$$

It is not hard to see that the above is correct, sound, and perfectly zero-knowledge for any verifier. As usual, this method can be made non-interactive by executing all repetitions in parallel and creating the challenge c by hashing all data before the second step.