Efficient Blind and Partially Blind Signatures Without Random Oracles

Tatsuaki Okamoto

NTT Laboratories, Nippon Telegraph and Telephone Corporation
1-1 Hikarino-oka, Yokosuka, 239-0847 Japan
okamoto.tatsuaki@lab.ntt.co.jp

Abstract. This paper proposes a new efficient signature scheme from bilinear maps that is secure in the standard model (i.e., without the random oracle model). Our signature scheme is more effective in many applications (e.g., blind signatures, group signatures, anonymous credentials etc.) than the existing secure signature schemes in the standard model such as the Boneh-Boyen [6], Camenisch-Lysyanskaya [10], Cramer-Shoup [15] and Waters [33] schemes (and their variants). The security proof of our scheme requires a slightly stronger assumption, the 2SDH assumption, than the SDH assumption used by Boneh-Boyen. As typical applications of our signature scheme, this paper presents efficient blind signatures and partially blind signatures that are secure in the standard model. Here, partially blind signatures are a generalization of blind signatures (i.e., blind signatures are a special case of partially blind signatures) and have many applications including electronic cash and voting. Our blind signature scheme is much more efficient than the existing secure blind signature schemes in the standard model such as the Camenisch-Koprowski-Warinsch [8] and Juels-Luby-Ostrovsky [22] schemes, and is also almost as efficient as the most efficient blind signature schemes whose security has been analyzed heuristically or in the random oracle model. Our partially blind signature scheme is the first one that is secure in the standard model and it is very efficient (almost as efficient as our blind signatures). We also present a blind signature scheme based on the Waters signature scheme.

1 Introduction

1.1 Background

Digital Signatures: The concept of digital signatures was invented by Diffie and Hellman [17], and their security was formalized by Goldwasser, Mical and Rivest [21]. A secure signature scheme exists if and only if a one-way function exists [26, 32]. However, the general solution is far from yielding any practical applications.

Using the random oracle model, much more efficient secure signature schemes have been presented such as RSA-FDH, RSA-PSS, Fiat-Shamir and Schnorr signature schemes. However, the random oracle model cannot be realized in the standard (plain) model. In addition, signatures with hash functions (random oracles) are less suitable to several applications (e.g., group signatures).

Several efficient schemes that are secure in the standard model have recently been presented. There are two classes of such schemes, ones are based on the strong RSA assumption (i.e., based on the integer factoring (IF) problem), while the others are based on bilinear maps (i.e., based on the discrete logarithm (DL) problem). The Camenisch-Lysyanskaya [10], Cramer-Shoup [15], Fischlin [19] and Gennaro-Halevi-Rabin [20] schemes are based on the strong RSA assumption. The Boneh-Boyen [6], Camenisch-Lysyanskaya [10], and Waters [33] schemes are based on bilinear maps.

Digital signatures not only provide basic signing functionality but also are important building blocks for many applications such as blind signatures (for electronic voting and electronic cash), group signatures and credentials. In the light of these applications, the schemes based on bilinear maps (i.e., based on the discrete logarithm problem) are better than those based on the strong RSA assumption (i.e., based on the integer factoring problem), since we can often more easily construct efficient protocols based on the DL problem (because the order of a DL-based group can be published but the order of an IF-based multiplicative group cannot), and the data size is shorter with bilinear maps than with IF problems.

Among the bilinear-map-based schemes, the Boneh-Boyen scheme is not suitable to many applications such as blind signatures and credentials, since the signature forms $\sigma \leftarrow g^{1/(x+m+sy)}$, where (x, y) is the secret key, m is a message and (σ, s) is the signature, so it is hard to separate an operation (blinding, encryption etc.) with m from another operation that uses the secret key.

The Waters scheme is better than the Boneh-Boyen scheme, since a message operation, through the form $\prod_{i \in \mathcal{M}} u_i$, can be separated from another operation that uses the secret key. However, as shown in Section 9 the protocol of proving the knowledge of a message is not so efficient.

Blind Signatures: Since the concept of blind signatures was introduced by Chaum [13], it has been used in numerous applications, most prominently in electronic voting and electronic cash. Informally, blind signatures allow a user to obtain signatures from a signer on any document in such a manner that the signer learns nothing about the message that is being signed. The security of blind signatures was formalized by [22, 28].

Even in the random oracle model, only a few secure blind signature schemes have been proposed [1, 4, 27–30]; [4] requires a non-standard strong assumption and [28–30] only allow a user to make a poly-logarithmically (not polynomially) bounded number of interactions with a signer, while [1, 27] are secure for a polynomially number of interactions.

Only two secure blind signature schemes have been presented in the standard model [8, 22]. However, the construction of [22] is based on a general two-party protocol and is thus extremely inefficient. The solution of [8] is much more efficient than that of [22], but it is still much less efficient than the secure blind signature schemes in the random oracle model [1, 4, 27–30]. For example, the protocol of [8] is much more complicated (where proofs of knowledge for at least 40 variables are required for a user) than that of [4, 28, 29], and requires many interactions between user and signer. Recently, a new blind signature scheme

that is concurrently secure without random oracles has been presented [23], but it is not in the standard model but in the common reference string (CRS) model.

Partially Blind Signatures: One particular shortcoming of the concept of blind signatures is that, since the singer's view of the message to be signed is completely blocked, the signer has no control over the attributes except for those bound by the public key. For example, a shortcoming can be seen in a simple electronic cash system where a bank issues a blind signature as an electronic coin. Since the bank cannot set the value on any blindly issued coin, it has to use different public keys for different coin values. Hence the shops and customers must always carry a list of those public keys in their electronic wallet, which is typically a smart card whose memory is very limited. Some electronic voting schemes also face the same problem.

A *partially* blind signature scheme allows the signer to explicitly include common information in the blind signature under some agreement with the receiver. This concept is a generalization of blind signatures since the (normal) blind signatures are a special case of *partially* blind signatures where the common information is a null string.

The notion of partially blind signatures was introduced in [2], and the formal security definition and a secure partially blind signature scheme in the random oracle model were presented by [3]. However, no partially blind signature scheme secure in the standard model has been proposed.

1.2 Our Result

This paper proposes new digital signatures, blind signatures, and partially blind signatures that are secure in the standard model:

- (Digital signatures:)
  We propose a new efficient signature scheme secure in the standard model that is more suitable to many applications than the existing signature schemes secure in the standard model [6, 10, 15, 33]. The security proof of our scheme requires a slightly stronger assumption, the 2SDH assumption, than the SDH assumption used by [6].
- (Blind signatures:)
  We propose a secure blind signature scheme in the standard model that is almost as efficient as the most efficient blind signature schemes whose security has been analyzed heuristically or in the random oracle model.
- (Partially blind signatures:)
  We propose the first secure partially blind signature scheme in the standard model. This scheme is almost as efficient as our blind signatures.

The proposed (partially) blind signature scheme is secure for polynomially many synchronized (or constant-depth concurrent) attacks, but not for general concurrent attacks. This paper presents an efficient way to convert our (partially) blind signature scheme in the standard model to a scheme secure for general concurrent attacks in the common reference string (CRS) model.

This paper also presents (partially) blind signatures from the Waters scheme that are secure in the standard model under the BDH assumption. The (partially) blind signatures are much less practical than the above-mentioned proposed scheme.

2 Preliminaries

2.1 Definition of Secure (Partially) Blind Signature Scheme

In this section we recall the definition of a secure *partially blind* signature scheme [3, 8]. Note that this definition includes that of a secure *blind* signature scheme [22] as a special case where the piece of information shared by the signer and user, info, is a null string, $\perp$ (i.e., $\mathsf{info} = \perp$).

Although our definition is based on [3, 8], our *blindness* definition is slightly stronger than [3, 8] as follows:

- Signer $\mathcal{S}^*$ can arbitrarily choose pk in ours, while pk must be honestly generated in [3, 8].
- Even if only one of two users, $\mathcal{U}_0$ or $\mathcal{U}_1$, outputs a valid signature, $\mathcal{S}^*$ is allowed to obtain the valid signature and output the decision, b', in our definition, while only when both users, $\mathcal{U}_0$ and $\mathcal{U}_1$, output valid signatures, $\mathcal{S}^*$ is allowed to obtain them in [3, 8].

Partially blind signature scheme: In the scenario of issuing a partially blind signature, the signer and the user are assumed to agree on a piece of common information, denoted as info. In some applications, info may be decided by the signer, while in other applications it may just be sent from the user to the signer. Anyway, this negotiation is done outside of the signature scheme, and we want the signature scheme to be secure regardless of the process of agreement.

Definition 1. *(Partially Blind Signature Scheme) A Partially blind signature scheme is made up of four (interactive) algorithms (machines)* $(\mathcal{G}, \mathcal{S}, \mathcal{U}, \mathcal{V})$.

- $\mathcal{G}$ *is a probabilistic polynomial-time algorithm that takes security parameter* n *and outputs a public and secret key pair* (pk, sk).
- $\mathcal{S}$ *and* $\mathcal{U}$ *are a pair of probabilistic interactive Turing machines each of which has a public input tape, a private input tape, a private random tape, a private work tape, a private output tape, a public output tape, and input and output communication tapes. The random tape and the input tapes are read-only, and the output tapes are write-only. The private work tape is read-write. The public input tape of* $\mathcal{U}$ *contains* pk *generated by* $\mathcal{G}(1^n)$ *and* info. *The public input tape of* $\mathcal{S}$ *contains* info. *The private input tape of* $\mathcal{S}$ *contains* sk, *and that for* $\mathcal{U}$ *contains message* m. $\mathcal{S}$ *and* $\mathcal{U}$ *engage in the signature issuing protocol and stop in polynomial-time in* n. *When they stop, the public output tape of* $\mathcal{S}$ *contains either* `completed` *or* `not-completed`. *Similarly, the private output tape of* $\mathcal{U}$ *contains either* $\perp$ *or* (m, σ).
- $\mathcal{V}$ *is a (probabilistic) polynomial-time algorithm that takes* $(pk, \mathsf{info}, m, \sigma)$ *and outputs either* `accept` *or* `reject`.

Definition 2. *(Completeness) If $\mathcal{S}$ and $\mathcal{U}$ follow the signature issuing protocol with common input (pk, info), then, with probability of at least $1 - 1/n^c$ for sufficiently large n and some constant c, $\mathcal{S}$ outputs* `completed`*, and $\mathcal{U}$ outputs (m, σ) that satisfies $\mathcal{V}(pk, \mathsf{info}, m, \sigma) =$* `accept`*. The probability is taken over the coin flips of $\mathcal{G}$, $\mathcal{S}$ and $\mathcal{U}$.*

We say message-signature tuple $(\mathsf{info}, m, \sigma)$ is *valid* with regard to pk if it leads $\mathcal{V}$ to `accept`.

Partial blindness: To define the blindness property, let us introduce the following game among adversarial signer $\mathcal{S}^*$ and two honest users $\mathcal{U}_0$ and $\mathcal{U}_1$.

1. Adversary $\mathcal{S}^*(1^n, \mathsf{info})$ outputs pk and (m_0, m_1).
2. Set up the input tapes of $\mathcal{U}_0, \mathcal{U}_1$ as follows:
   - Randomly select $b \in \{0, 1\}$ and put m_b and $m_{\bar{b}}$ on the private input tapes of $\mathcal{U}_0$ and $\mathcal{U}_1$, respectively ($\bar{b}$ denotes $1 - b$ hereafter).
   - Put (info, pk) on the public input tapes of $\mathcal{U}_0$ and $\mathcal{U}_1$.
   - Randomly select the contents of the private random tapes.
3. Adversary $\mathcal{S}^*$ engages in the signature issuing protocol with $\mathcal{U}_0$ and $\mathcal{U}_1$.
4. If $\mathcal{U}_0$ and $\mathcal{U}_1$ output valid signatures $(\mathsf{info}, m_b, \sigma_b)$ and $(\mathsf{info}, m_{\bar{b}}, \sigma_{\bar{b}})$, respectively, then give those outputs to $\mathcal{S}^*$ in random order. If either $\mathcal{U}_0$ or $\mathcal{U}_1$ outputs a valid signature, $(\mathsf{info}, m_b, \sigma_b)$ or $(\mathsf{info}, m_{\bar{b}}, \sigma_{\bar{b}})$, then give this output to $\mathcal{S}^*$. Give $\perp$ to $\mathcal{S}^*$ otherwise.
5. $\mathcal{S}^*$ outputs $b' \in \{0, 1\}$.

We define

$$\mathtt{Adv}^{\mathrm{blind}}_{\mathrm{PBS}} = 2 \cdot \Pr[b' = b] - 1,$$

where the probability is taken over the coin tosses made by $\mathcal{S}^*, \mathcal{U}_0$ and $\mathcal{U}_1$.

Definition 3. *(Partial Blindness) Adversary $\mathcal{S}^*$ (t, ϵ)-breaks the blindness of a partially blind signature scheme if $\mathcal{S}^*$ runs in time at most t, and $\mathtt{Adv}^{\mathrm{blind}}_{\mathrm{PBS}}$ is at least ϵ. A partially blind signature scheme is (t, ϵ)-blind if no adversary $\mathcal{S}^*$ (t, ϵ)-breaks the blindness of the scheme.*

Remark: (Partially Perfect Blindness) As usual, one can go for a stronger notion of blindness depending on the power of the adversary and its success probability. A scheme provides partially *perfect* blindness if it is $(\infty, 0)$-blind.

Unforgeability: To define unforgeability, let us introduce the following game among adversarial user $\mathcal{U}^*$ and an honest signer $\mathcal{S}$.

1. (pk, sk) is generated by $\mathcal{G}(1^n)$, pk is put on the public input tapes of $\mathcal{U}^*$ and $\mathcal{S}$, and sk is put on the private input tape of $\mathcal{S}$.
2. For each run of the signature issuing protocol with $\mathcal{S}$, adversary $\mathcal{U}^*$ outputs info, which is put on the public input tape of $\mathcal{S}$. Then, $\mathcal{U}^*$ engages in the signature issuing protocol with $\mathcal{S}$ in a concurrent and interleaving way.

3. For each info, let ℓ_{info} be the number of executions of the signature issuing protocol where $\mathcal{S}$ outputs `completed`, given info on its input tape. (For info that has never appeared on the input tape of $\mathcal{S}$, define $\ell_{\mathsf{info}} = 0$.) Even when $\mathsf{info} = \perp$, $\ell_\perp$ is also defined in the same manner.
4. $\mathcal{U}^*$ wins the game if $\mathcal{U}^*$ output ℓ valid signatures $(\mathsf{info}, m_1, \sigma_1), \ldots, (\mathsf{info}, m_\ell, \sigma_\ell)$ for some info such that
   (a) $m_i \neq m_j$ for any pair (i, j) with $i \neq j$ $(i, j \in \{1, \ldots, \ell\})$.
   (b) $\ell > \ell_{\mathsf{info}}$.

We define $\mathtt{Adv}_{\mathrm{PBS}}^{\mathrm{unforge}}$ to be the probability that $\mathcal{U}^*$ wins the above game, taken over the coin tosses made by $\mathcal{U}^*$, $\mathcal{G}$ and $\mathcal{S}$.

Definition 4. *(Unforgeability) An adversary $\mathcal{U}^*$ (t, q_S, ϵ)-forges a partially blind signature scheme if $\mathcal{U}^*$ runs in time at most t, $\mathcal{U}^*$ executes at most q_S times the signature issuing protocol, and $\mathtt{Adv}_{\mathrm{PBS}}^{\mathrm{unforge}}$ is at least ϵ. A partially blind signature scheme is (t, q_S, ϵ)-unforgeable if no adversary $\mathcal{U}^*$ (t, q_S, ϵ)-forges the scheme.*

2.2 Bilinear Groups

This paper follows the notation regarding bilinear groups in [7, 6]. Let $(\mathbb{G}_1, \mathbb{G}_2)$ be bilinear groups as follows:

1. $\mathbb{G}_1$ and $\mathbb{G}_2$ are two cyclic groups of prime order p, where possibly $\mathbb{G}_1 = \mathbb{G}_2$,
2. g_1 is a generator of $\mathbb{G}_1$ and g_2 is a generator of $\mathbb{G}_2$,
3. ψ is an isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$, with $\psi(g_2) = g_1$,
4. e is a non-degenerate bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, where $|\mathbb{G}_1| = |\mathbb{G}_2| = |\mathbb{G}_T| = p$, i.e.,
   (a) Bilinear: for all $u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}$, $e(u^a, v^b) = e(u, v)^{ab}$,
   (b) Non-degenerate: $e(g_1, g_2) \neq 1$ (i.e., $e(g_1, g_2)$ is a generator of $\mathbb{G}_T$),
5. e, ψ and the group action in $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ can be computed efficiently.

3 Assumptions

Here we introduce a new assumption, the 2-variable strong Diffie-Hellman (2SDH) assumption on which the security of the proposed signature scheme is based.

***q* 2-Variable Strong Diffie-Hellman (*q*-2SDH) Problem:** Let $(\mathbb{G}_1, \mathbb{G}_2)$ be bilinear groups shown in Section 2.2. The q-2SDH problem in $(\mathbb{G}_1, \mathbb{G}_2)$ is defined as follows: given a $(2q+6)$-tuple $(g_1, g_2, g_2^x, \ldots, g_2^{x^q}, g_2^y, g_2^{yx}, \ldots, g_2^{yx^q}, g_2^{\frac{y+b}{x+a}}, a, b)$ as input, output pair $(g_1^{\frac{1}{x+c}}, c)$ where $c \in \mathbb{Z}_p^*$. Algorithm $\mathcal{A}$ has advantage, $\mathtt{Adv}_{2SDH}(q)$, in solving q-2SDH in $(\mathbb{G}_1, \mathbb{G}_2)$ if

$$\mathtt{Adv}_{2SDH}(q) \leftarrow \Pr[\, \mathcal{A}(g_1, g_2, g_2^x, \ldots, g_2^{x^q}, g_2^y, g_2^{yx}, \ldots, g_2^{yx^q}, g_2^{\frac{y+b}{x+a}}, a, b) = (g_1^{\frac{1}{x+c}}, c) \,],$$

where the probability is taken over the random choices of $g_2 \in \mathbb{G}_2$, $x, y, a, b \in \mathbb{Z}_p^*$, and the coin tosses of $\mathcal{A}$.

Definition 5. *Adversary $\mathcal{A}$ (t, ϵ)-breaks the q-2SDH problem if $\mathcal{A}$ runs in time at most t and $\mathtt{Adv}_{2SDH}(q)$ is at least ϵ. The (q, t, ϵ)-2SDH assumption holds if no adversary $\mathcal{A}$ (t, ϵ)-breaks the q-2SDH problem.*

Variant of q 2-Variable Strong Diffie-Hellman (q-2SDH$_S$) Problem: The q-2SDH$_S$ problem in $(\mathbb{G}_1, \mathbb{G}_2)$ is defined as follows: given a $(3q+4)$-tuple $(g_1, g_2, g_2^x, g_2^y, g_2^{\frac{y+b_1}{x+a_1}}, \ldots, g_2^{\frac{y+b_q}{x+a_q}}, g_2^{a_1}, \ldots, g_2^{a_q}, b_1, \ldots, b_q)$ as input, output a pair $(g_1^{\frac{y+d}{x+c}}, g_2^c, d)$ where $b_1, \ldots, b_q, d \in \mathbb{Z}_p^*$ and $d \notin \{b_1, \ldots, b_q\}$. Algorithm $\mathcal{A}$ has advantage, $\mathtt{Adv}_{2SDH_S}(q)$, in solving q-2SDH$_S$ in $(\mathbb{G}_1, \mathbb{G}_2)$ if

$$\mathtt{Adv}_{2SDH_S}(q)$$
$$\leftarrow \Pr[\, \mathcal{A}(g_1, g_2, g_2^x, g_2^y, g_2^{\frac{y+b_1}{x+a_1}}, \ldots, g_2^{\frac{y+b_q}{x+a_q}}, g_2^{a_1}, \ldots, g_2^{a_q}, b_1, \ldots, b_q) = (g_1^{\frac{y+d}{x+c}}, g_2^c, d)\,],$$

where $b_1, \ldots, b_q, d \in \mathbb{Z}_p^*$ and $d \notin \{b_1, \ldots, b_q\}$, and the probability is taken over the random choices of $g_2 \in \mathbb{G}_2$, $x, y, a_1, b_1, \ldots, a_q, b_q \in \mathbb{Z}_p^*$, and the coin tosses of $\mathcal{A}$.

Definition 6. *Adversary $\mathcal{A}$ (t, ϵ)-breaks the q-2SDH$_S$ problem if $\mathcal{A}$ runs in time at most t and $\mathtt{Adv}_{2SDH_S}(q)$ is at least ϵ. The (q, t, ϵ)-2SDH$_S$ assumption holds if no adversary $\mathcal{A}$ (t, ϵ)-breaks the q-2SDH$_S$ problem.*

Remark 1: We occasionally drop t and ϵ and refer to the q-2SDH (or q-2SDH$_S$) assumption rather than the (q, t, ϵ)-2SDH (or (q, t, ϵ)-2SDH$_S$) assumption. We also sometimes drop q- and $_S$ and refer to the 2SDH assumption rather than the q-2SDH or q-2SDH$_S$ assumption.

Remark 2: (Relation between the 2SDH and 2SDH$_S$ assumptions)
The 2SDH and 2SDH$_S$ assumptions are closely related in a manner similar to the equivalence of $(q-1)$-wDHA assumption and q-CAA assumption [25], where the q-wDHA problem is to output $g_1^{\frac{1}{x}}$, given a $(q+2)$-tuple $(g_1, g_2, g_2^x, \ldots, g_2^{x^q})$ as input, and the q-CAA problem is to output pair $(g_1^{\frac{1}{x+c}}, c)$ where $c \in \mathbb{Z}_p^*$ and $c \notin \{a_1, \ldots, a_q\}$, given a $(2q+3)$-tuple $(g_1, g_2, g_2^x, g_2^{\frac{1}{x+a_1}}, \ldots, g_2^{\frac{1}{x+a_q}}, a_1, \ldots, a_q)$ as input.

4 The Proposed Signature Scheme

This section presents the proposed secure signature scheme in the standard model under the 2SDH assumption.

Let $(\mathbb{G}_1, \mathbb{G}_2)$ be bilinear groups as shown in Section 2.2. Here, we assume that the message, m, to be signed is an element in $\mathbb{Z}_p^*$, but the domain can be extended to all of $\{0,1\}^*$ by using a collision resistant hash function $H : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$, as mentioned in Section 3.5 in [6].

4.1 Signature Scheme

Key generation: Randomly select generators $g_2, u_2, v_2 \in \mathbb{G}_2$ and set $g_1 \leftarrow \psi(g_2)$, $u_1 \leftarrow \psi(u_2)$, and $v_1 \leftarrow \psi(v_2)$. Randomly select $x \in \mathbb{Z}_p^*$ and compute $w_2 \leftarrow g_2^x \in \mathbb{G}_2$. The public and secret keys are:

Public key: g_1, g_2, w_2, u_2, v_2
Secret key: x

Signature generation: Let $m \in \mathbb{Z}_p^*$ be the message to be signed. Signer $\mathcal{S}$ randomly selects r and s from $\mathbb{Z}_p^*$, and computes

$$\sigma \leftarrow (g_1^m u_1 v_1^s)^{1/(x+r)}.$$

Here $1/(x+r) \bmod p$ (and $m/(x+r) \bmod p$ and $s/(x+r) \bmod p$) are computed. In the unlikely event that $x+r \equiv 0 \mod p$, we try again with a different random r. (σ, r, s) is the signature of m.

Signature verification: Given public-key $(g_1, g_2, w_2, u_2, v_2)$, message m, and signature (σ, r, s), check that $m, r, s \in \mathbb{Z}_p^*$, $\sigma \in \mathbb{G}_1$, $\sigma \neq 1$, and

$$e(\sigma, w_2 g_2^r) = e(g_1, g_2^m u_2 v_2^s).$$

If they hold, the verification result is `valid`; otherwise the result is `invalid`.

Remark: Here we assume that $g_1 = \psi(g_2)$ has been confirmed when the public-key is registered. Alternatively, $g_1 = \psi(g_2)$ can be confirmed in the signature verification procedure, or g_1 is not included in the public-key and $g_1 = \psi(g_2)$ is calculated in the signature verification process.

4.2 A Performance Improvement Technique (Precomputation)

By introducing additional secret key $y, z \in \mathbb{Z}_p^*$ such that $u_2 = g_2^y$ and $v_2 = g_2^z$, we can apply a precomputation technique for signature generation.

Before getting message m, signer $\mathcal{S}$ randomly selects r, δ from $\mathbb{Z}_p^*$, and computes $\sigma \leftarrow g_1^{\delta/(x+r)}$ as the precomputation of a signature. Given message m, $\mathcal{S}$ computes s such that $s \leftarrow (\delta - m - y)/z \bmod p$, where $1/z \bmod p$ can be also precomputed.

4.3 Security

Theorem 1. *If the $(q_S + 1, t', \epsilon')$-2SDH assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$, the proposed signature scheme is (t, q_S, ϵ)-strongly-existentially-unforgeable against adaptive chosen message attacks, provided that*

$$\epsilon \geq 3 q_S \epsilon', \quad \textit{and} \quad t \leq t' - \Theta(q_S^2 T),$$

where T is the maximum time for a single exponentiation in $\mathbb{G}_1$ and $\mathbb{G}_2$.

Proof. (Sketch) Assume $\mathcal{A}$ is an adversary that (t, q_S, ϵ)-forges the signature scheme. We will then construct algorithm $\mathcal{B}$ that breaks the $(q_S + 1)$-2SDH assumption with (t', ϵ'). Hereafter, we often use $q \leftarrow q_S + 1$ (as well as q_S).

An informal outline of our proof is as follows: First we classify the output (forgery) of $\mathcal{A}$ into three types (Types-1,2,3). We will then show that any type of output allows $\mathcal{B}$ to break the q-2SDH assumption. Type-1 forgery leads to breaking the q-SDH (to which q-2SDH is reducible) assumption in a manner similar to that in [6]. Type-2 forgery leads to breaking the q-2SDH assumption

by producing $g_2^{\frac{1}{x+b}}$ from the q-2SDH problem including $g_2^{\frac{y+a}{x+b}}$. Type-3 forgery leads to breaking the discrete logarithm (to which q-2SDH is reducible).

First, we introduce three types of forgers, $\mathcal{A}$. Let $(g_1, g_2, w_2, u_2, v_2)$ be given to $\mathcal{A}$ as a public-key, and $z \leftarrow \log_{g_2} v_2 \in \mathbb{Z}_p^*$ (i.e., $v_2 = g_2^z$). Suppose $\mathcal{A}$ asks for signatures on messages $m_1, \ldots, m_{q_S} \in \mathbb{Z}_p^*$ and is given signatures (σ_i, r_i, s_i) for $i = 1, \ldots, q_S$ on these messages. The three types of forgers are as follows:

Type-1 forger outputs forged signature $(m*, \sigma^*, r^*, s^*)$ such that $r^* \notin \{r_1, r_2, \ldots, r_{q_S}\}$.

Type-2 forger outputs forged signature $(m^*, \sigma^*, r^*, s^*)$ such that $r^* \in \{r_1, r_2, \ldots, r_{q_S}\}$ (i.e., $r^* = r_k$ for some $k \in \{1, \ldots, q_S\}$) and $m^* + s^* z \not\equiv m_k + s_k z \pmod{p}$.

Type-3 forger outputs forged signature $(m*, \sigma^*, r^*, s^*)$ such that $r_1^* \in \{r_1, r_2, \ldots, r_{q_S}\}$ (i.e., $r^* = r_k$ for some $k \in \{1, \ldots, q_S\}$) and $m^* + s^* z \equiv m_k + s_k z \pmod{p}$. Note that in this case $s^* \neq s_k$, since $s^* = s_k$ implies $m^* = m_k$ and $\sigma^* = \sigma_k$.

Algorithm $\mathcal{B}$ is constructed as follows:

1. (Input:)
   $(g_1, A_0, A_1, \ldots, A_q, B_0, B_1, \ldots, B_q, C, a, b)$, where $A_i = g_2^{x^i}$, $B_i = g_2^{yx^i}$, and $C = g_2^{\frac{y+b}{x+a}}$ $(i = 0, 1, \ldots, q)$.
2. (Coin flip:)
   Algorithm $\mathcal{B}$ first picks a random value $c_{\text{type}} \in \{1, 2, 3\}$ that indicates its guess for the type of forger that $\mathcal{A}$ will emulate. The subsequent actions performed by $\mathcal{B}$ differ with $c_{\text{type}} \in \{1, 2, 3\}$ as follows:
3. (If $c_{\text{type}} = 1$;)
   In this case, q-SDH assumption is broken in a manner similar to that shown in [6].
4. (If $c_{\text{type}} = 2$;)
   (a) (Key setup)
   $\mathcal{B}$ randomly selects $z, r_i (\neq a)$ $(i = 1, \ldots, q-1)$ from $\mathbb{Z}_p^*$. Let $f(X) \leftarrow \prod_{i=1}^{q-1}(X + r_i) \bmod p = \sum_{i=0}^{q-1} \beta_i X^i$. $\mathcal{B}$ can efficiently calculate $\beta_i \in \mathbb{Z}_p^*$ $(i = 0, \ldots, q-1)$ from r_i $(i = 1, \ldots, q-1)$.
   $\mathcal{B}$ computes

$$g_2' \leftarrow \prod_{i=0}^{q-1} A_i^{\beta_i} = g_2^{f(x)}, \quad w_2' \leftarrow \prod_{i=0}^{q-1} A_{i+1}^{\beta_i} = (g_2')^x,$$

$$u_2' \leftarrow \prod_{i=0}^{q-1} B_i^{\beta_i} = (g_2')^y, \quad v_2' \leftarrow (g_2')^z.$$

   Let $g_1' \leftarrow \psi(g_2')$, $u_1' \leftarrow \psi(u_2')$ and $v_1' \leftarrow \psi(v_2')$.
   $\mathcal{B}$ gives $(g_1', g_2', w_2', u_2', v_2')$ to $\mathcal{A}$ as a public-key of the signature scheme.
   (b) (Simulation of signing oracle)
   Upon receiving a query to the signing oracle, $\mathcal{B}$ simulates the reply to $\mathcal{A}$ as follows:

Let $f_i(X) \leftarrow f(X)/(X+r_i) \bmod p = \prod_{j=1, j\neq i}^{q-1}(X+r_i) \bmod p = \sum_{j=0}^{q-2} \gamma_j X^j$.
$\mathcal{B}$ can efficiently calculate $\gamma_j \in \mathbb{Z}_p^*$ $(j = 0, \ldots, q-2)$ from r_l $(l \neq i \wedge l = 1, \ldots, q-1)$.
First, $\mathcal{B}$ randomly selects $k \in \{1, 2, \ldots, q-1\}$.
For each query $i \in \{1, 2, \ldots, k-1, k+1, q-1\}$ (i.e., $i \neq k$) with message m_i from $\mathcal{A}$ to the signing oracle, $\mathcal{B}$ randomly selects $s_i \in \mathbb{Z}_p^*$, and computes

$$\sigma_i \leftarrow \Big(\prod_{j=0}^{q-2} \psi(A_j)^{\gamma_j}\Big)^{m_i + s_i z} \Big(\prod_{j=0}^{q-2} \psi(B_j)^{\gamma_j}\Big) = (g_1')^{(m_i + y + s_i z)/(x + r_i)}.$$

$\mathcal{B}$ returns (σ_i, r_i, s_i) to $\mathcal{A}$ as the reply to the query. Clearly this is a valid signature for public-key $(g_1', g_2', w_2', u_2', v_2')$.
For the k-th query with message m_k from $\mathcal{A}$ to the signing oracle, $\mathcal{B}$ computes $\omega_i, d \in \mathbb{Z}_p^*$ $(i = 1, \ldots, q-2)$ such that $f(X) = c(X)(X + a) + d \bmod p$, $c(X) \leftarrow \sum_{i=0}^{q-2} \omega_i X^i$ and $d \in \mathbb{Z}_p^*$, and computes

$$\sigma_k \leftarrow \psi(C)^d \Big(\prod_{i=0}^{q-2} \psi(A_i)^{\omega_i}\Big)^b \prod_{i=0}^{q-2} \psi(B_i)^{\omega_i} = (g_1')^{(m_k + y + s_k z)/(x + r_k)},$$
$$s_k \leftarrow (b - m_k)/z \bmod p, \qquad r_k \leftarrow a.$$

$\mathcal{B}$ returns (σ_k, r_k, s_k) to $\mathcal{A}$ as the reply to the query.

(c) (Output) When $\mathcal{A}$ outputs a (valid) forgery $(m^*, \sigma^*, r^*, s^*)$, $\mathcal{B}$ checks whether $r^* = a$ and $m^* + s^* z \not\equiv m_k + s_k z \pmod p$. If $r^* \neq a$ or $m^* + s^* z \equiv m_k + s_k z \pmod p$, then $\mathcal{B}$ outputs `failure` and aborts. Otherwise, $m^* + s^* z \not\equiv m_k + s_k z \pmod p$. Let $b^* \leftarrow m^* + s^* z \bmod p$. (Here $b = m_k + s_k z \bmod p$). Since $b^* \neq b$, $\mathcal{B}$ can compute

$$\eta \leftarrow \left(\frac{(\sigma^*/\sigma_k)^{1/(b^* - b)}}{\prod_{i=0}^{q-2} \psi(A_i)^{\omega_i}}\right)^{1/d} = g_1^{1/(x+a)}.$$

$\mathcal{B}$ outputs (η, a).

5. (If $c_{\mathrm{type}} = 3$;)

(a) (Key setup)
$\mathcal{B}$ randomly selects x', y' from $\mathbb{Z}_p^*$.
$\mathcal{B}$ computes

$$g_2' \leftarrow A_0 = g_2, \quad w_2' \leftarrow (g_2')^{x'}, \quad u_2' \leftarrow (g_2')^{y'}, \quad v_2' \leftarrow A_1 = g_2^x.$$

Here we rename x as z' just for representation, so

$$v_2' = (g_2')^{z'}.$$

Let $g_1' \leftarrow g_1$.
$\mathcal{B}$ gives $(g_1', g_2', w_2', u_2', v_2')$ to $\mathcal{A}$ as a public-key of the signature scheme.

(b) (Simulation of signing oracle) Since $\mathcal{B}$ knows x', the simulation of the signing oracle exactly replicates the signing oracle.

(c) (Output) When $\mathcal{A}$ outputs a (valid) forgery $(m^*, \sigma^*, r^*, s^*)$, $\mathcal{B}$ checks whether $r^* \in \{r_1, \ldots, r_{q_S}\}$ (i.e., $r^* = r_k$ for some $k \in \{1, \ldots, q_S\}$) and $s^* \neq s_k$. If $r^* \notin \{r_1, \ldots, r_{q_S}\}$ or $s^* \neq s_k$, then $\mathcal{B}$ outputs `failure` and aborts. Otherwise, $\mathcal{B}$ computes

$$z^* \leftarrow (m_k - m^*)/(s^* - s_k) \bmod p,$$

and checks whether $A_1 = A_0^{z^*}$. If it holds, $z^* = z' = x$. $\mathcal{B}$ then randomly selects $c \in \mathbb{Z}_p^*$ and can compute $\eta \leftarrow g_1^{1/(z^*+c)} = g_1^{1/(x+c)}$.
$\mathcal{B}$ outputs (η, c).

Since the value of c_{type} is independent from the type of forgery, $\mathcal{B}$ breaks the q-2SDH assumption with probability at least $\epsilon/(3q_S)$. $\dashv$

5 Variant of the Proposed Signature Scheme

This section presents a slight variant of the proposed signature scheme presented in the previous section. This variant is used by our blind signatures.

5.1 Signature Scheme

The variant scheme is the same as the proposed signature scheme except for the signature generation and verification parts as follows: in this variant, the signature is

$$(\sigma \leftarrow (g_1^m u_1 v_1^s)^{1/(x+r)}, \alpha \leftarrow g_2^r, s),$$

while in the proposed signature scheme in Section 4, the signature is (σ, r, s).

The signature verification equation of this variant is

$$e(\sigma, w_2 \alpha) = e(g_1, g_2^m u_2 v_2^s),$$

while the proposed signature scheme in Section 4, the signature verification equation is $e(\sigma, w_2 g_2^r) = e(g_1, g_2^m u_2 v_2^s)$.

5.2 Security

Theorem 2. *If the (q_S, t', ϵ')-2SDH$_S$ assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$, the proposed signature scheme is (t, q_S, ϵ)-existentially-unforgeable against adaptive chosen message attacks, provided that*

$$\epsilon \geq 2\epsilon', \quad \textit{and} \quad t \leq t' - O(q_S T),$$

where T is the maximum time for a single exponentiation in $\mathbb{G}_1$ and $\mathbb{G}_2$.

The proof is shown in the full paper version.

6 The Proposed (Partially) Blind Signature Scheme

This section shows the proposed *partially blind* signature scheme, which includes our *blind* signature scheme as a special case where $m_0 = 0$ or $h_2 = 1$.

6.1 Partially Blind Signature Scheme

Let $(\mathbb{G}_1, \mathbb{G}_2)$ be bilinear groups as shown in Section 2.2. Here, we also assume that the messages, m_0 and m_1, to be (partially blindly) signed are elements in $\mathbb{Z}_p^*$, but the domain can be extended to all of $\{0,1\}^*$ by using a collision resistant hash function $H : \{0,1\}^* \to \mathbb{Z}_p^*$, as mentioned in Section 3.5 in [6].

Key generation: Randomly select generators $g_2, u_2, v_2, h_2 \in \mathbb{G}_2$ and set $g_1 \leftarrow \psi(g_2)$, $u_1 \leftarrow \psi(u_2)$, $v_1 \leftarrow \psi(v_2)$, and $h_1 \leftarrow \psi(h_2)$. Randomly select $x \in \mathbb{Z}_p^*$ and compute $w_2 \leftarrow g_2^x \in \mathbb{G}_2$. The public and secret keys are:

Public key: $g_1, g_2, w_2, u_2, v_2, h_2$
Secret key: x

Partially blind signature generation:

1. Signer S and user U agree on common information m_0 (which is info in Section 2.1) in an predetermined way.
2. U randomly selects $s, t \in \mathbb{Z}_p^*$, computes

$$X \leftarrow h_1^{m_0 t} g_1^{m_1 t} u_1^t v_1^{st},$$

and sends X to S. Here, m_1 is the message to be blindly signed along with common information m_0. In addition, U proves to S that U knows $(t, m_1 t, t, st)$ for $X = (h_1^{m_0})^t g_1^{m_1 t} u_1^t v_1^{st}$ using the witness indistinguishable proof as follows:

(a) U randomly selects a_1, a_2, a_3 from $\mathbb{Z}_p^*$, computes

$$W \leftarrow (h_1^{m_0})^{a_2} g_1^{a_1} u_1^{a_2} v_1^{a_3},$$

and sends W to S.

(b) S randomly selects $\eta \in \mathbb{Z}_p^*$ and sends η to U.

(c) U computes

$$b_1 \leftarrow a_1 + \eta m_1 t \bmod p, \quad b_2 \leftarrow a_2 + \eta t \bmod p, \quad b_3 \leftarrow a_3 + \eta st \bmod p,$$

and sends (b_1, b_2, b_3) to S.

(d) S checks whether the following equation holds or not:

$$(h_1^{m_0})^{b_2} g_1^{b_1} u_1^{b_2} v_1^{b_3} = W X^{\eta}.$$

If it holds, S accepts. Otherwise, S rejects and aborts.

3. If S accepts the above protocol, S randomly selects $r \in \mathbb{Z}_p^*$. In the unlikely event that $x + r \equiv 0 \mod p$, S tries again with a different random r. S also randomly selects $\ell \in \mathbb{Z}_p^*$, computes

$$Y \leftarrow (X v_1^{\ell})^{1/(x+r)} \quad \text{and} \quad R \leftarrow g_2^r,$$

and sends (Y, R, ℓ) to U.
Here, $Y = (X v_1^{\ell})^{1/(x+r)} = (h_1^{m_0} g_1^{m_1} u_1 v_1^{s+\ell/t})^{t/(x+r)}$.
4. U randomly selects $f \in \mathbb{Z}_p^*$, and computes

$$\tau = (ft)^{-1} \mod p, \quad \sigma \leftarrow Y^{\tau}, \quad \alpha \leftarrow w_2^{f-1} R^f, \quad \beta \leftarrow s + \ell/t \mod p.$$

Here, $\sigma = (h_1^{m_0} g_1^{m_1} u_1 v_1^{s+\ell/t})^{1/(fx+fr)} = (h_1^{m_0} g_1^{m_1} u_1 v_1^{s+\ell/t})^{1/(x+(f-1)x+fr)} = (h_1^{m_0} g_1^{m_1} u_1 v_1^{\beta})^{1/(x+\delta)}$, and $\alpha = w_2^{f-1} R^f = g_2^{(f-1)x+fr} = g_2^{\delta}$, where $\delta = (f-1)x + fr \mod p$.
5. (σ, α, β) is the partially blind signature of (m_0, m_1), where m_0 is common information between S and U, and m_1 is blinded to S.

Signature verification: Given public-key $(g_1, g_2, w_2, u_2, v_2, h_2)$, common information m_0, message m_1, and signature (σ, α, β), check that $m_0 \in \mathbb{Z}_p^*$, $m_1 \in \mathbb{Z}_p^*$, $\beta \in \mathbb{Z}_p$, $\sigma \neq 1$, $\sigma \in \mathbb{G}_1$, $\alpha \in \mathbb{G}_2$, and

$$e(\sigma, w_2 \alpha) = e(g_1, h_2^{m_0} g_2^{m_1} u_2 v_2^{\beta}).$$

6.2 Security

Theorem 3. *The proposed blind signature scheme ($m_0 = 0$ or $h_2 = 1$) is perfectly blind.*

Proof. Even if dishonest signer $\mathcal{S}^*$ outputs any public-key, $(g_2, w_2, u_2, v_2) \in (\mathbb{G}_2)^4$ and $g_1 = \psi(g_2)$, the view of $\mathcal{S}^*$, $(X, W, \eta, b_1, b_2, b_3)$ as well as $\mathcal{S}$'s randomness, in the signature generation protocol is perfectly (information theoretically) independent from the value of (m, s, f), since $X = (g_1^m u_1 v_1^s)^t$ is perfectly independent from (m, s), the protocol is witness indistinguishable with respect to (m, s) against any dishonest $\mathcal{S}^*$, and f is not used in the protocol with $\mathcal{S}^*$.

Hence, the value of (m, δ, β) is perfectly independent from the view of $\mathcal{S}^*$, where $\delta = (x+r)f - x \mod p$ and $\beta \leftarrow s + \ell/t \mod p$. Here, $\sigma = (g_1^m u_1 v_1^{\beta})^{1/(x+\delta)}$, $\alpha = g_2^{\delta}$, and (σ, α, β) is the (blind) signature of m. Therefore, the signature along with m, $(m, \sigma, \alpha, \beta)$, is also perfectly independent from the view of $\mathcal{S}^*$, since σ and α are perfectly dependent on (m, δ, β). ⊣

Definition 7. *Let suppose a protocol between two parities, Alice and Bob. In a round of the protocol, Alice and Bob exchange messages, $a, b, c, \ldots, d$, where the first move is Alice (i.e., Alice sends a and Bob returns b etc.). We now consider q rounds of the protocol execution. Here $(a_i, b_i, c_i, \ldots, d_i)$ is the exchanged messages in the i-th round ($i = 1, \ldots, q$). We say that a protocol between Alice and Bob is executed in a synchronized run of q rounds of the protocol, if the q*

rounds of the protocol consists of L sequential intervals and each interval, or the j-th interval ($j = 1, \ldots, L$), consists of the parallel run of q_j ($q_j \in \{1, \ldots, q\}$ rounds of the protocol. $q = q_1 + \cdots + q_L$. Therefore, the first interval consists of: the first move from Alice is $(a_1, a_2, \ldots, a_{q_1})$, the second move from Bob is $(b_1, a_2, \ldots, b_{q_1})$, and so on. After completing the first interval, the second interval starts and consists of: the first move from Alice is $(a_{q_1+1}, a_{q_1+2}, \ldots, a_{q_1+q_2})$, the second move from Bob is $(b_{q_1+1}, b_{q_1+2}, \ldots, b_{q_1+q_2})$, and so on.

Clearly the synchronized run is a generalization of the parallel and sequential runs.

Theorem 4. *If the (q_S, t', ϵ')-2SDH$_S$ assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$, the proposed blind signature scheme ($m_0 = 0$ or $h_2 = 1$) is (t, q_S, ϵ)-unforgeable against an L-interval synchronized run of adversaries, provided that*

$$\epsilon' \leq \frac{1 - 1/(L+1)}{16} \cdot \epsilon, \quad \textit{and} \quad t' \geq \frac{24L \log(L+1)}{\epsilon} \cdot (t + \Theta(T)) + \Theta(q_S T),$$

where T is the maximum time for a single exponentiation in $\mathbb{G}_1$ and $\mathbb{G}_2$.

Proof. (Sketch)

Assume $\mathcal{A}$ is an adversary that (t, q_S, ϵ)-forges the blind signature scheme. We will then construct an algorithm $\mathcal{B}$ that (t'', q_S, ϵ'')-forges the proposed signature scheme (basic signature scheme) presented in Section 5. This leads to an algorithm that breaks the 2SDH$_S$ assumption with $(q_S, t'' + O(q_S T), \epsilon''/2)$ by Theorem 2.

$\mathcal{B}$, given $(g_1, g_2, w_2, u_2, v_2)$ as a public key of the basic signature scheme, provides them to $\mathcal{A}$ as a public key for blind signatures.

$\mathcal{B}$ is allowed to access the signing oracle of the basic signature scheme q_S times. By using this signing oracle, $\mathcal{B}$ plays the role of an honest signer against $\mathcal{A}$ (dishonest user).

First, $\mathcal{A}$ requests $\mathcal{B}$ to sign X along with the witness indistinguishable (WI) protocol on witness $(mt \bmod p, t, st \bmod p)$ against $\mathcal{B}$'s random challenge $\eta \in \mathbb{Z}_p^*$. After completing the WI protocol, $\mathcal{B}$ resets $\mathcal{A}$ to the initial state of the WI protocol and runs the same procedure with the same commitment value of W and another random challenge $\eta' \in \mathbb{Z}_p^*$ ($\eta \neq \eta'$). If $\mathcal{B}$ succeeds in completing the WI protocol twice with different challenges η and η' such that

$$g_1^{b_1} u_1^{b_2} v_1^{b_3} = W X^{\eta}, \quad g_1^{b'_1} u_1^{b'_2} v_1^{b'_3} = W X^{\eta'}, \tag{1}$$

$\mathcal{B}$ can compute

$$\begin{aligned} m' &\leftarrow (b_1 - b'_1)/(\eta - \eta') \bmod p, \\ t &\leftarrow (b_2 - b'_2)/(\eta - \eta') \bmod p, \\ s' &\leftarrow (b_3 - b'_3)/(\eta - \eta') \bmod p, \end{aligned} \tag{2}$$

such that

$$X = g_1^{m'} u_1^{t} v_1^{s'}.$$

$\mathcal{B}$ computes

$$m \leftarrow m'/t \bmod p, \quad s \leftarrow s'/t \bmod p. \tag{3}$$

$\mathcal{B}$ then resumes the protocol just after the WI protocol, and sends m to the signing oracle. The signing oracle returns to $\mathcal{B}$ (σ, α, β) such that $(\sigma \leftarrow g_1^m u_1 v_1^{\beta})^{1/(x+r)}$ and $\alpha = g_2^r$. $\mathcal{B}$ computes

$$Y \leftarrow \sigma^t, \quad \ell \leftarrow t(\beta - s) \bmod p, \tag{4}$$

and returns $\mathcal{A}$ (Y, ℓ).

$\mathcal{B}$ repeats the above procedures (at the request of $\mathcal{A}$) q_S times. If all q_S rounds of the above procedures are completed, $\mathcal{A}$ finally outputs at least $q_S + 1$ valid signatures with distinct messages. From the pigeon-hole principle, among at least $q_S + 1$ distinct messages with valid signatures that $\mathcal{A}$ outputs, at least one message with valid signature is different from the q_S messages with valid signatures given by the signing oracle. This contradicts the q_S-unforgeability of the basic signature scheme.

The remaining problem in this strategy is how to execute all q_S rounds of the WI protocol twice with distinct challenges η and η' in a synchronized run with $\mathcal{A}$.

Claim. $\mathcal{B}$ can execute all q_S rounds of the WI protocol twice with distinct challenges η and η' in a synchronized run with $\mathcal{A}$ with probability at least $(1 - 1/(L+1))\epsilon/8$ under the condition that $\mathcal{B}$ rewinds $\mathcal{A}$ with random challenges at most $24L \log(L+1)/\epsilon$ times in total (or in L intervals).

Combining this result with Theorem 2 we obtain this theorem. ⊣

Theorem 5. *The proposed partially blind signature scheme is perfectly blind.*

The proof is almost the same as that in Theorem 3.

Theorem 6. *If the (q_S, t', ϵ')-2SDH$_S$ assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$, the proposed partially blind signature scheme is (t, q_S, ϵ)-unforgeable against an L-interval synchronized run of adversaries, provided that*

$$\epsilon' \leq \frac{1 - 1/(L+1)}{32} \cdot \epsilon, \quad \textit{and} \quad t' \geq \frac{48L \log(L+1)}{\epsilon} \cdot (t + \Theta(T)) + \Theta(q_S T),$$

where T is the maximum time for a single exponentiation in $\mathbb{G}_1$ and $\mathbb{G}_2$.

Remark: (Constant-depth concurrency) We can define a specific type of concurrent runs, *constant-depth concurrent* runs, in which, informally speaking, only a constant depth of purely inner rounds is allowed in all paths. Synchronized runs are a specific type of depth-1 concurrent runs. We can show that our blind signature scheme is still secure against a constant-depth concurrent run of adversaries under the same assumption and model. The result is presented in the full paper version.

6.3 Generalization

(m_0, m_1) with an additional key h_2 is generalized to $(m_0, \ldots, m_l)$ with additional key $(h_{2,1}, \ldots, h_{2,l})$. Arbitrary subset in $\{m_0, \ldots, m_l\}$ can be blinded messages and the remaining be common messages.

7 Conversion to Fully Concurrent Security in the CRS Model

As mentioned above, the proposed (partially) blind signature scheme is secure against a synchronized run of adversaries (or more generally, a constant-depth concurrent run of adversaries). In this section, we show how to convert the proposed scheme to a scheme secure against a fully-concurrent run of adversaries. Our proposed blind signature scheme is secure in the *plain model* (without any setup assumptions), while the converted scheme is secure in the *common reference string (CRS) model*. The key idea is similar to [23], and uses the Paillier encryption for a simulator to extract blind messages with the help of the CRS model, and also uses a trapdoor commitment [16] to realize a concurrent zero-knowledge protocol. For simplicity of description, we will show a blind signature scheme, but it is straightforward to extend it to our partially blind signature scheme.

Key generation: Randomly select generators $g_2, u_2, v_2 \in \mathbb{G}_2$ and set $g_1 \leftarrow \psi(g_2)$, $u_1 \leftarrow \psi(u_2)$, and $v_1 \leftarrow \psi(v_2)$. Randomly select $x \in \mathbb{Z}_p^*$, and compute $w_2 \leftarrow g_2^x \in \mathbb{G}_2$. In addition, randomly select secret and public keys of the Paillier encryption, two prime integers P and Q, and $(N = PQ, G)$, where $|N| = (6 + 3c_0)|p|$ (c_0 is a constant and $0 < c_0 < 1$). The public and secret keys, (pk, sk), of a trapdoor commitment, *commit*, [16] are also generated.

The public and secret keys and CRS are:

Public key: g_1, g_2, w_2, u_2, v_2
Secret key: x
CRS: N, G, pk
Trapdoor of CRS: P, Q, sk

Blind signature generation:

1. U checks whether $g_2, w_2, u_2, v_2 \in \mathbb{G}_2$ and $g_1 = \psi(g_2)$. If they hold, U proceeds the following signature generation protocol.
2. U randomly selects $s, t \in \mathbb{Z}_p^*$ and $A \in \mathbb{Z}_{N^2}$, computes

$$X \leftarrow g_1^{mt} u_1^t v_1^{st}, \quad D \leftarrow G^{(mt \bmod p) + t2^K + (st \bmod p)2^{2K}} A^N \bmod N^2,$$

and sends (X, D) to S. Here $K = (2+c_0)|p|$, and $m \in \mathbb{Z}_p^*$ is the message to be blindly signed. In addition, U proves to S that U knows $(mt \bmod p, t, st \bmod p)$ for X as follows:

(a) U randomly selects a_1, a_2, a_3 from $\{0,1\}^{(2+c_1)|p|}$ (c_1 is a constant and $0 < c_1 < c_0 < 1$), $B \in \mathbb{Z}_{N^2}$ and r^* from the domain, computes

$$W \leftarrow g_1^{a_1} u_1^{a_2} v_1^{a_3}, \quad E \leftarrow G^{a_1 + a_2 2^K + a_3 2^{2K}} B^N \bmod N^2,$$

$$C \leftarrow commit(E, r^*, pk),$$

and sends (W,C) to S.

(b) S randomly selects $\eta \in \mathbb{Z}_p^*$ and sends η to U.

(c) U computes

$$b_1 \leftarrow a_1 + \eta(mt \bmod p), \quad b_2 \leftarrow a_2 + \eta t, \quad b_3 \leftarrow a_3 + \eta(st \bmod p),$$

$$F \leftarrow BA^\eta \bmod N^2,$$

and sends (b_1, b_2, b_3, F) as well as (E, r^*) to S.

(d) S checks whether the following equation holds or not:

$$|b_i| \leq (2 + c_1)|p| \quad (i = 1, 2, 3), \qquad C = commit(E, r^*, pk).$$

$$g_1^{b_1} u_1^{b_2} v_1^{b_3} = WX^\eta, \qquad G^{b_1 + b_2 2^K + b_3 2^{2K}} F^N \equiv ED^\eta \pmod{N^2}$$

If it holds, S accepts. Otherwise, S rejects and aborts.

3. The remaining procedure is the same as that of the original blind signature scheme.

Signature verification: Same as that of the original blind signature scheme.

Security: The signature generation protocol is statistically WI except D, which is the Paillier encryption of a message. Since the Paillier encryption is semantically secure under the N-th residue assumption, this blind signature scheme satisfies blindness under this assumption.

If the WI protocol in the signature generation protocol is accepted by signer, simulator can extract (m, s, t) by decrypting D without rewinding $\mathcal{A}$ with high probability, by using the trapdoor of CRS (i.e., P, Q). That is, this scheme is unforgeable against any concurrent run of adversaries under the 2SDH_S assumption. (The proof is shown in the full paper version.)

8 Other Applications

We have shown the application of the proposed signature scheme to blind and partially blind signatures. The proposed signature scheme also supports other applications such as restrictive (partially) blind signatures, group signatures [24], verifiably encrypted signatures, anonymous credentials and chameleon hash signatures. (The full paper version presents restrictive (partially) blind signatures based on our (partially) blind signatures.)

9 (Partially) Blind Signatures from the Waters Scheme

9.1 The Proposed Blind Signature Scheme from the Waters Scheme

Key generation: Let a symmetric bilinear group, $(\mathbb{G}_1, \mathbb{G}_1)$, be used in this scheme. Randomly select $\alpha \in \mathbb{Z}_p^*$. Randomly select generators $g, g_2, u', u_1, \ldots, u_n \in \mathbb{G}_1$ and set $g_1 \leftarrow g^\alpha$.

Public key: $g, g_1, g_2, u', u_1, \ldots, u_n$
Secret key: g_2^α

Blind signature generation: Let m be the n-bit message to be signed, m_i the ith bit of m.

1. User U randomly selects $t \in \mathbb{Z}_p^*$, computes

$$X \leftarrow (u' \prod_{i=1}^{n} u_i^{m_i})^t,$$

   and sends X to S. In addition, U proves to S that U knows $(t, m_1, \ldots, m_n)$ with $m_i \in \{0,1\}$ for $X = (u' \prod_{i=1}^{n} u_i^{m_i})^t$ using the witness indistinguishable Σ protocols. For example,
   (a) U randomly selects $\delta_1, \ldots, \delta_n \in \mathbb{Z}_p^*$, computes $M_i = u_i^{m_i}(u')^{\delta_i}$ $(i = 1, \ldots, n)$, and sends $(M_1, \ldots, M_n)$ to S.
   (b) U proves to S that U knows δ_i such that $M_i = (u')^{\delta_i}$ or $M_i = u_i(u')^{\delta_i}$ $(i = 1, \ldots, n)$. Such an OR-proof can be efficiently realized by a Σ protocol [3].
   (c) U proves to S that U knows $(t, \beta, \gamma_1, \ldots, \gamma_n)$ such that $X = (\prod_{i=1}^{n} M_i)^t (u')^\beta$, and $X = (u')^t \prod_{i=1}^{n} u_i^{\gamma_i}$, where $\beta \leftarrow t - t(\sum_{i=1}^{n} \delta_i) \bmod p$ and $\gamma_i \leftarrow t m_i$.
2. If S accepts the above protocol, S randomly selects $r \in \mathbb{Z}_p^*$, computes

$$Y_1 \leftarrow g_2^\alpha X^r, \quad Y_2 \leftarrow g^r,$$

   and sends (Y_1, Y_2) to U.
3. U randomly selects $s \in \mathbb{Z}_p^*$, and computes

$$\sigma_1 \leftarrow Y_1 (u' \prod_{i=1}^{n} u_i^{m_i})^s, \quad \sigma_2 \leftarrow Y_2^t g^s$$

4. $\sigma \leftarrow (\sigma_1, \sigma_2)$ is a blind signature.

Signature verification: Given public-key $(g, g_1, g_2, u', u_1, \ldots, u_n)$, message $m \in \mathbb{Z}_p^*$, and signature $\sigma = (\sigma_1, \sigma_2)$, check

$$e(\sigma_1, g) / e(\sigma_2, u' \prod_{i=1}^{n} u_i^{m_i}) = e(g_1, g_2).$$

If it holds, the verification result is `valid`; otherwise the result is `invalid`.

Remark: If adversary $\mathcal{A}$ executes in a synchronized (or constant-depth concurrent) run with simulator $\mathcal{B}$ (as signer), $\mathcal{B}$ can effectively extract $(m_1, \ldots, m_n)$ and t from $\mathcal{A}$. $\mathcal{B}$ can then reduce the basic Waters signature scheme attack to the proposed blind signature scheme attack. It is straightforward to realize a partially blind signature scheme in a similar manner. The major problem in the efficiency of the signing process is in proving the knowledge of many ($O(n)$) variables in the WI Σ protocols.

Acknowledgements

The author would like to thank anonymous reviewers of TCC 2006 for their invaluable comments and suggestions.

References

1. Abe, M., A Secure Three-Move Blind Signature Scheme for Polynomially Many Signatures, Eurocrypt'01, LNCS 2045, pp.136-151, Springer-Verlag (2001).
2. Abe, M. and Fujisaki, E., How to Date Blind Signatures, Asiacrypt'96, LNCS 1163, pp.244-251, Springer-Verlag (1996).
3. Abe, M. and Okamoto, T., Provably Secure Partially Blind Signatures, Crypto'00, LNCS 1880, pp.271-286, Springer-Verlag (2000).
4. Bellare, M., Namprempre, C., Pointcheval, D. and Semanko, M., The power of RSA inversion oracles and the security of Chaum's RSA-based blind signature scheme, Financial Cryptography'01, LNCS, Springer-Verlag (2001).
5. Boldyreva, A., Threshold Signature, Multisignature and Blind Signature Schemes Based on the Gap-Diffie-Hellman-Group Signature Scheme, PKC'03, LNCS 2567, pp.31-46, Springer-Verlag (2003).
6. Boneh, D. and Boyen, X., Short Signatures Without Random Oracles, Crypto'04, LNCS, Springer-Verlag (2004).
7. Boneh, D., Lynn, B. and Shacham, H., Short Signatures from the Weil Pairing, Asiacrypt'01, LNCS, Springer-Verlag (2001).
8. Camenisch, J., Koprowski, M. and Warinschi, B., Efficient Blind Signatures without Random Oracles, Forth Conference on Security in Communication Networks - SCN '04, LNCS, Springer-Verlag (2004).
9. Camenisch, J. and Lysyanskaya, A., Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation, Eurocrypt'01, LNCS 2045, pp. 93-118, Springer-Verlag (2001).
10. Camenisch, J. and Lysyanskaya, A., A signature scheme with efficient protocols, Security in communication networks, LNCS 2576, pp.268-289, Springer-Verlag (2002).
11. Camenisch, J. and Shoup, V., Practical verifiable encryption and decryption of discrete logarithms, Crypto'03, LNCS, pp. 126-144. Springer-Verlag (2003).
12. Camenisch, J. and Lysyanskaya, A., Signature Schemes and Anonymous Credentials from Bilinear Maps, Crypto'04, LNCS, Springer-Verlag (2004)
13. Chaum, D., Blind signatures for untraceable payments, Crypto'82, pp. 199-203. Plenum Press (1983).

14. Chow, S., Hui, L., Yiu, S. and Chow, K., Two Improved Partially Blind Signature Schemes from Bilinear Pairings, IACR Cryptology ePrint Archive, 2004/108 (2004).
15. Cramer, R. and Shoup, V., Signature schemes based on the strong RSA assumption, 6th ACM CCS, pp. 46-52. ACM press (1999).
16. Damgård, I., Efficient Concurrent Zero-Knowledge in the Auxiliary String Model, Eurocrypt'00, LNCS 1807, pp.418-430, Springer-Verlag (2000).
17. Diffie, W. and Hellma, M.E., New directions in cryptography, IEEE Trans. on Information Theory, IT-22(6), pp.644-654 (1976).
18. Fiat, A. and Shamir, A., How to prove yourself: Practical solution to identification and signature problems, Crypto'86, LNCS 263, Springer-Verlag (1987).
19. Fischlin, M., The Cramer-Shoup strong-RSA signature scheme revisited, PKC 2003, LNCS 2567, Springer-Verlag (2003).
20. Gennaro, R., Halevi, S. and Rabin, T., Secure hash-and-sign signatures without the random oracle, Eurocrypt'99, LNCS 1592, pp.123-139, Springer-Verlag (1999).
21. Goldwasser, S., Micali, S., and Rivest, R., A digital signature scheme secure against adaptive chosen-message attacks, SIAM Journal on Computing, 17, 2, pp.281-308 (1988).
22. Juels, A., Luby, M. and Ostrovsky, R., Security of blind digital signatures, Crypto'97, LNCS 1294, pp. 150-164, Springer-Verlag (1997).
23. Kiayias, A. and Zhou, H., Two-Round Concurrent Blind Signatures without Random Oracles, IACR Cryptology ePrint Archive, 2005/435 (2005)
24. Makita, T., Manabe, Y. and Okamoto, T., Short Group Signatures with Efficient Flexible Join, Manuscript (2005).
25. Mitsunari, S., Sakai, R. and Kasahara, M., A New Traitor Tracing, IEICE Trans. E-85-A, 2, pp. 481-484 (2002).
26. Naor, M. and Yung, M., Universal one-way hash functions and their cryptographic applications, 21st STOC, pp. 33-43, ACM (1989).
27. Pointcheval, D., Strengthened security for blind signatures, Eurocrypt'98, LNCS, pp.391-405, Springer-Verlag (1998).
28. Pointcheval, D. and Stern, J., Provably secure blind signature schemes, Asiacrypt'96, LNCS, Springer-Verlag (1996).
29. Pointcheval, D. and Stern, J., New blind signatures equivalent to factorization, ACM CCS, pp. 92-99. ACM Press (1997).
30. Pointcheval, D. and Stern, J., Security arguments for digital signatures and blind signatures, Journal of Cryptology, 13, 3, pp.361-396, Springer-Verlag (2000).
31. Schnorr, C.P., Security of Blind Discrete Log Signatures against Interactive Attacks, ICICS'01, LNCS 2229, pp.1-12, Springer-Verlag (2001).
32. Rompel, J., One-way functions are necessary and sufficient for secure signatures, STOC, pp.387-394, ACM (1990).
33. Waters, B., Efficient Identity-Based Encryption Without Random Oracles, Eurocrypt'05, LNCS 3494, pp. 114-127, Springer-Verlag (2005).
34. Zhang, F., Safavi-Naini, R. and Susilo, W, Efficient Verifiably Encrypted Signature and Partially Blind Signature from Bilinear Pairings, Indocrypt'03, LNCS 2904, pp. 191-204, Springer-Verlag (2003). Revised version available at http://www.uow.edu.au/susilo.