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
March 17, 2006

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. 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 more efficient than the existing secure blind signature schemes in the standard model such as the Camenisch-Koprowski-Warinsch [9] and Juels-Luby-Ostrovsky [24] schemes. Our partially blind signature scheme is the first one that is secure in the standard model and it is also efficient (as efficient as our blind signatures). The security proof of our blind and partially blind signature schemes requires the 2SDH assumption, a stronger variant of the SDH assumption introduced by Boneh and Boyen [7]. This paper also presents an efficient way to convert our (partially) blind signature scheme in the standard model to a scheme secure for a concurrent run of users in the common reference string (CRS) model. Finally, we 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 [18], and their security was formalized by Goldwasser, Mical and Rivest [23]. A secure signature scheme exists if and only if a one-way function exists [28, 36]. 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 for 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, some 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 [11], Cramer-Shoup [16], Fischlin [20] and Gennaro-Halevi-Rabin [22] schemes are based on the strong RSA assumption. The Boneh-Boyen [7], Camenisch-Lysyanskaya [11], and Waters [37] 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 for 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 form $\prod_{i \in \mathcal{M}} u_i$, can be separated from another operation that uses the secret key. However, as shown in Section 10 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 [14], 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 [24, 31].

Even in the random oracle model, only a few secure blind signature schemes have been proposed [2, 5, 30–33]; [5] requires a non-standard strong assumption and [31–33] only allow a user to make a poly-logarithmically (not polynomially) bounded number of interactions with a signer, while [2, 30] are secure for a polynomially number of interactions.

Only two secure blind signature schemes have been presented in the standard model [9, 24]. However, the construction of [24] is based on a general two-party protocol and is thus extremely inefficient. The solution of [9] is much more efficient than that of [24], but it is still much less efficient than the secure blind signature schemes in the random oracle model [2, 5, 30–33]. For example, the protocol of [9] is much more complicated (where proofs of knowledge for at least 40 variables are required for a user) than that of [5, 31, 32], and requires many interactions between user and signer. Recently, a new blind signature scheme that is concurrently secure without random oracles has been presented [25], but it is in the common reference string (CRS) model, not in the standard model.

Partially Blind Signatures: One particular shortcoming of the concept of blind signatures is that, since the signer'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 [3], and the formal security definition and a secure partially blind signature scheme in the random oracle model were presented by [4]. 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:

- (Digital signatures:)
  We propose a new efficient signature scheme secure in the standard model that is more suitable for many applications including blind signatures than the existing signature schemes secure in the standard model [7, 11, 16, 37]. The security of our basic signature scheme depends on the strong Diffie-Hellman (SDH) assumption introduced by [7], and its variant signature scheme depends on the 2SDH assumption, a variant of the SDH assumption.
- (Blind signatures:)
  We propose a secure blind signature scheme in the standard model whose efficiency is comparable to that of existing blind signature schemes whose security has been analyzed heuristically or in the random oracle model. The security proof of our scheme depends on the 2SDH assumption.
- (Partially blind signatures:)
  We propose the first secure partially blind signature scheme in the standard model. This scheme is as efficient as our blind signatures and secure under the 2SDH assumptions.
- (Conversion to concurrent security:)
  The proposed (partially) blind signature scheme is secure for polynomially many synchronized (or constant-depth concurrent) runs of users, but the security proof does not work for general concurrent runs. This paper presents an efficient way to convert our (partially) blind signature scheme in the standard model to a scheme secure for a general concurrent run of users in the common reference string (CRS) model.
- (Blind signatures from Waters:)
  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 Signature Schemes

In this section we recall the standard notion of security, existential unforgeability (against chosen message attacks) [23] as well as a slightly stronger notion of security for a signature scheme, *strong* existential unforgeability (against chosen message attacks) [7].

A signature scheme is made up of three algorithms (machines) $(\mathcal{G}, \mathcal{S}, \mathcal{V})$, for key generation, signing and verification. Here we omit the description of these algorithms except the minimum syntax description, $\mathcal{G}(1^n) = (pk, sk)$, $\mathcal{S}(sk, m) = \sigma$ and $\mathcal{V}(pk, m, \sigma) =$ `accept/reject`. For a detailed description, see [23]).

Existential unforgeability: To define existential unforgeability, we introduce the following game among adversary $\mathcal{A}$ and honest signer $\mathcal{S}$.

1. (Key setup:)
   Run key generation algorithm $\mathcal{G}(1^n)$ to obtain a pair of public-key and secret-key, (pk, sk). pk is given to adversary $\mathcal{A}$, and (pk, sk) is given to signer $\mathcal{S}$.
2. (Queries to signing oracle:)
   $\mathcal{A}$ adaptively requests $\mathcal{S}$ (or signing oracle) to sign on at most q_S messages of his choice $m_1, \ldots, m_{q_S}$. $\mathcal{S}$ responds to m_i with a signature $\sigma_i = \mathcal{S}(sk, m_i)$.
3. (Output:)
   Eventually, $\mathcal{A}$ outputs pair (m, σ). $\mathcal{A}$ wins the game if
   (a) m is not any of m_i $(i = 1, \ldots, q_S)$.
   (b) $\mathcal{V}(pk, m, \sigma) = \mathtt{accept}$.

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

Definition 1. *(Existential Unforgeability) Adversary $\mathcal{A}$ (t, q_S, ϵ)-forges a signature scheme if $\mathcal{A}$ runs in time at most t, $\mathcal{A}$ makes at most q_S queries to $\mathcal{S}$, and $\mathtt{Adv}_{\mathrm{Sig}}^{\mathrm{unforge}}$ is at least ϵ. A signature scheme is (t, q_S, ϵ)-existentially-unforgeable under adaptive chosen message attacks if no adversary $\mathcal{A}$ (t, q_S, ϵ)-forges the scheme.*

Remark: (Strong Existential Unforgeability) If the condition in Step 3a in the above game is changed to "(m, σ) is not any of (m_i, σ_i)" (instead that "m is not any of m_i") $(i = 1, \ldots, q_S)$, we obtain a stronger notion of unforgeability. If a scheme satisfies the above definition of unforgeability under this stronger notion, we say that it is (t, q_S, ϵ)-*strongly*-existentially-unforgeable under adaptive chosen message attacks.

2.2 Definition of Secure (Partially) Blind Signature Scheme

In this section we recall the definition of a secure *partially blind* signature scheme [4, 9]. Note that this definition includes that of a secure *blind* signature scheme [24] 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 [4, 9], our *blindness* definition is slightly stronger than [4, 9] as follows [1] :

- Signer $\mathcal{S}^*$ can arbitrarily choose pk in ours, while pk must be honestly generated in [4, 9].

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

[1] The stronger definition is also introduced in [1]. The definition shown in our preliminary version [29] has a trivial flaw in the blindness definition, where 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. Then, if $\mathcal{S}^*$ interacts incorrectly with $\mathcal{U}_0$ and correctly with $\mathcal{U}_1$ in the signing protocol, $\mathcal{S}^*$ can identify m_b (i.e., b) for $\mathcal{U}_0$ by checking $\mathcal{U}_1$'s valid signature.

- *$\mathcal{G}$ 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 3. *(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 (σ_0, σ_1) 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 4. *(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 $\ell_{\mathsf{info}} = 0$.) When $\mathsf{info} = \perp$, $\ell_\perp$ is also defined in the same manner.
4. $\mathcal{U}^*$ wins the game if $\mathcal{U}^*$ outputs 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 $\mathtt{Adv}^{\mathrm{unforge}}_{\mathrm{PBS}}$ 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 5. *(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 $\mathtt{Adv}^{\mathrm{unforge}}_{\mathrm{PBS}}$ 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.*

2.3 Bilinear Groups

This paper follows the notation regarding bilinear groups in [8,7]. 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 first recall the definition of the strong Diffie-Hellman (SDH) assumption introduced in [7], on which the security of our signature scheme is based, and then introduce new assumptions, the 2-variable strong Diffie-Hellman (2SDH), on which the security of a variant of our signature scheme and the proposed (partially) blind signature scheme is based.

3.1 Strong Diffie-Hellman (SDH) Assumption:

Let $(\mathbb{G}_1, \mathbb{G}_2)$ be bilinear groups as shown in Section 2.3. The q-SDH problem in $(\mathbb{G}_1, \mathbb{G}_2)$ is defined as follows: given the $(q+2)$-tuple $(g_1, g_2, g_2^x, \ldots, g_2^{x^q})$ as input, output pair $(g_1^{\frac{1}{x+c}}, c)$ where $c \in \mathbb{Z}_p^*$. Algorithm $\mathcal{A}$ has advantage, $\mathtt{Adv}_{\mathrm{SDH}}(q)$, in solving q-SDH in $(\mathbb{G}_1, \mathbb{G}_2)$ if

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

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

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

3.2 2-Variable Strong Diffie-Hellman (2SDH) Assumption:

The q-2SDH problem in $(\mathbb{G}_1, \mathbb{G}_2)$ is defined as follows: given a $(3q+4)$-tuple $(g_1, g_2, w_2 \leftarrow g_2^x, u_2 \leftarrow g_2^y, g_2^{\frac{y+b_1}{x+a_1}}, \ldots, g_2^{\frac{y+b_q}{x+a_q}}, a_1, \ldots, a_q, b_1, \ldots, b_q)$ as input, output $(\sigma \leftarrow g_1^{\frac{y+d}{\theta x+\rho}}, \alpha \leftarrow g_2^{\theta x+\rho}, d)$ as well as $\mathtt{Test}(\alpha) \leftarrow (U, V)$, where $a_1, \ldots, a_q, b_1, \ldots, b_q, d, \theta, \rho \in \mathbb{Z}_p^*$; $w_1 \leftarrow \psi(w_2), \sigma, U \in \mathbb{G}_1$; $\alpha, V \in \mathbb{G}_2$; and

$$e(\sigma, \alpha) = e(g_1, u_2 g_2^d), \quad e(U, \alpha) = e(w_1, w_2) \cdot e(g_1, V), \quad d \notin \{b_1, \ldots, b_q\}. \tag{1}$$

Algorithm $\mathcal{A}$ has advantage, $\mathtt{Adv}_{\mathrm{2SDH}}(q)$, in solving q-2SDH in $(\mathbb{G}_1, \mathbb{G}_2)$ if

$$\begin{aligned} &\mathtt{Adv}_{\mathrm{2SDH}}(q) \\ &\leftarrow \Pr[\, \mathcal{A}(g_1, g_2, w_2, u_2, g_2^{\frac{y+b_1}{x+a_1}}, \ldots, g_2^{\frac{y+b_q}{x+a_q}}, a_1, \ldots, a_q, b_1, \ldots, b_q) \\ &= (\sigma, \alpha, d, \mathtt{Test}(\alpha)), \end{aligned} \tag{2}$$

where Eq. (1) holds. 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 7. *Adversary $\mathcal{A}$ (t, ϵ)-breaks the q-2SDH problem if $\mathcal{A}$ runs in time at most t and $\mathtt{Adv}_{\mathrm{2SDH}}(q)$ is at least ϵ. The (q, t, ϵ)-2SDH assumption holds if no adversary $\mathcal{A}$ (t, ϵ)-breaks the q-2SDH problem.*

Lemma 1. *Assume that each variable X of $\{\alpha, \mathtt{Test}(\alpha)\}$ in $\mathbb{G}_i (i = 1, 2)$ corresponds to a unique index value, $(\xi_X, \zeta_X) \in (\mathbb{Z}_p^*)^2$, such that $X = w_i^{\xi_X} g_i^{\zeta_X}$. For two variables, X and Y, we also assume that $e(X, Y)$ corresponds to a unique index value, $(\omega_1, \omega_2, \omega_3) \in (\mathbb{Z}_p^*)^4$, such that $e(X, Y) = e(w_1, w_2)^{\omega_1} \cdot e(g_1, w_2)^{\omega_2} \cdot e(g_1, g_2)^{\omega_3}$. Here, $(\omega_1, \omega_2, \omega_3)$ is consistent with (ξ_X, ζ_X) and (ξ_Y, ζ_Y) with respect to the pairing operation, and the index of a variable is also consistent with other indexes of other variables with respect to the group operation in $\mathbb{G}_1$ and $\mathbb{G}_2$.*

Then, there exists $\mathtt{Test}(\alpha)$ satisfying Eq.(1), if and only if α corresponds to (θ, ρ) with $\theta \neq 0$ such that $\alpha = w_2^{\theta} g_2^{\rho}$.

Remark: To break the assumption in this lemma (i.e., to find index values, (ξ_X, ζ_X)'s, of X, or index values, $(\omega_1, \omega_2, \omega_3)$'s, of $e(X, Y)$) implies to compute x (in Eq.(2)) efficiently. Therefore, if adversary $\mathcal{A}$ cannot correctly compute X, Y satisfying the form $e(X, Y) = c$ without knowing (extracting) the value of $(\xi_X, \zeta_X, \xi_Y, \zeta_Y)$, then $\mathcal{A}$ can output (θ, ρ) satisfying $\alpha = w_2^{\theta} g_2^{\rho}$ with $\theta \not\equiv 0 \pmod{p}$, if $\mathcal{A}$ provides a valid answer of the 2SDH problem, assuming that $\mathcal{A}$ cannot compute x. Then, $\mathcal{A}$ can break the 2SDH$^+$ problem to be introduced in Remark 3 below.

Proof. (If part:)

Let $X \in \{\alpha, U, V\}$ and $X = w_i^{\xi_X} g_i^{\zeta_X}$. For two variables, $X = w_1^{\xi_X} g_1^{\zeta_X}$ and $Y = w_2^{\xi_Y} g_2^{\zeta_Y}$, of $\{\alpha, \mathtt{Test}(\alpha)\}$, if $e(X,Y) = e(w_1,w_2)^{\omega_1} e(g_1,w_2)^{\omega_2} e(g_1,g_2)^{\omega_3}$ holds, then $\omega_1 = \xi_X\xi_Y, \omega_2 = \xi_X\zeta_Y + \zeta_X\xi_Y, \omega_3 = \zeta_X\zeta_Y$.

Suppose $\mathtt{Test}(\alpha)$ holds. From $e(U,\alpha) = e(w_1,w_2)e(g_1,V)$,

$$\begin{aligned} e(U,\alpha) &= e(w_1^{\xi_U} g_1^{\zeta_U}, w_2^{\xi_\alpha} g_2^{\zeta_\alpha}) = e(w_1,w_2)^{\xi_U\xi_\alpha} e(g_1,w_2)^{\xi_U\zeta_\alpha + \zeta_U\xi_\alpha} e(g_1,g_2)^{\zeta_U\zeta_\alpha} \\ &= e(w_1,w_2) e(g_1,w_2)^{\xi_V} e(g_1,g_2)^{\zeta_V}. \end{aligned}$$

Therefore,

$$\xi_U\xi_\alpha \equiv 1 \pmod p, \quad \xi_U\zeta_\alpha + \zeta_U\xi_\alpha \equiv \xi_V \pmod p, \quad \zeta_U\zeta_\alpha \equiv \zeta_V \pmod p.$$

Then, $\xi_\alpha \not\equiv 0 \pmod p$.

(Only if part:)

If $\alpha \leftarrow w_2^\theta g_2^\rho$ $(\theta \neq 0)$, (U,V) satisfying Eq.(1) can be computed such that $U \leftarrow w_1^{1/\theta} g_1^\lambda$, $V \leftarrow w_2^{\theta\lambda + \rho/\theta} g_2^{\rho\lambda}$, where λ is randomly selected from $\mathbb{Z}_p^*$. ⊣

Lemma 2. *Given (α, U, V) with $e(U,\alpha) = e(w_1,w_2)\cdot e(g_1,V)$, for any value of $\theta \in \mathbb{Z}_p^*$ there exists $\rho \in \mathbb{Z}_p^*$ such that $\alpha = g_2^{\theta x + \rho}$.*

Proof. Given (α, U, V) with $e(U,\alpha) = e(w_1,w_2)\cdot e(g_1,V)$, for any value of $\theta \in \mathbb{Z}_p^*$ there exists $(\rho,\lambda) \in (\mathbb{Z}_p^*)^2$ such that

$$\log_{g_2}\alpha = \theta x + \rho \bmod p, \quad \log_{g_2} U = x/\theta + \lambda \bmod p, \quad \log_{g_2} V = (\theta\lambda + \rho/\theta)x + \rho\lambda.$$

⊣

Remark 1: The check for $\mathtt{Test}(\alpha)$ is introduced to avoid a trivial attack [35] such that randomly select $\pi \in \mathbb{Z}_p^*$, compute $\alpha \leftarrow g_2^\pi$, where α corresponds to g^{x+c}, and $g_1^{\frac{y+d}{x+c}} = (u_1 g_1^d)^{1/\pi}$ for an arbitrary $d \in \mathbb{Z}_p^*$, and output $(g_1^{\frac{y+d}{x+c}}, \alpha, d)$. Clearly this is a valid output of the 2SDH problem if Eq.(1) is not tested. The output in this trivial attack, where $\theta = 0$ with $\alpha = w_2^\theta g_2^\rho$, should be rejected by testing Eq.(1), as shown in Proposition 1.

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

The 2SDH assumption without explicit quantities (q,t,ϵ) denotes the (q,t,ϵ)-2SDH assumption with polynomial-time quantities for (q,t,ϵ), i.e., a polynomial number of q, polynomial-time of t, and negligible probability of ϵ in security parameter n.

Remark 3: (Relation between the SDH and 2SDH assumptions)

We now consider a variant of the 2SDH problem, the 2SDH$^+$ problem, as follows: given a $(3q+4)$-tuple $(g_1, g_2, g_2^x, g_2^y, g_2^{\frac{y+b_1}{x+a_1}}, \ldots, g_2^{\frac{y+b_q}{x+a_q}}, a_1, \ldots, a_q, b_1, \ldots, b_q)$ as input, output a pair $(g_1^{\frac{y+d}{x+c}}, c, d)$, where $(c,d) \neq (a_i, b_i)$ for all $i = 1, \ldots, q$. We can then define the 2SDH$^+$ assumption.

Clearly the 2SDH assumptions is stronger than the 2SDH$^+$ assumptions.

The 2SDH$^+$ assumption and the SDH assumption can be reduced to each other in a manner similar to the reduction in Theorem 1 ($c_{\mathrm{type}} = 1$ and 2) as well as the equivalence of $(q-1)$-wDHA assumption and q-CAA assumption [27], where the q-wDHA problem is to compute $g_1^{\frac{1}{x}}$, 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 compute 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 SDH assumption [2]

Let $(\mathbb{G}_1, \mathbb{G}_2)$ be bilinear groups as shown in Section 2.3. 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 $H : \{0,1\}^* \to \mathbb{Z}_p^*$, as mentioned in Section 3.5 in [7].

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 Performance

Signature length: A signature contains three elements (σ, r, s), each of which is around $|p|$ bits, so the signature length is around $3|p|$ bits. It is 1.5 times longer than that of the Boneh-Boyen scheme [7]. For example, if we use an elliptic curve described in [8], and $|p| = 250$, the signature length is 750 bits, which is still less than that of RSA based signatures with the same level of security.

[2] In our preliminary version [29], a variant of the 2SDH assumption, which is equivalent to the SDH assumption, was used to prove the security of this scheme. This signature scheme was implicitly introduced in the group signature scheme [21].

Computational complexity: Key and signature generation times are comparable to those of Boneh-Boyen [7] and BLS [8]. Signature verification time is comparable to that of [8], but around twice as slow as that of [7].

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 the 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 $1/z \bmod p$ and $(\delta - y)/z \bmod p$ can be also precomputed.

4.3 Security

Theorem 1. *If the (q_S+1, t', ϵ')-SDH 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. 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)-SDH 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-SDH assumption. Type-1 forgery leads to breaking the q-SDH assumption in a manner similar to that in [7]. Type-2 forgery leads to breaking the q-SDH assumption. Type-3 forgery leads to breaking the discrete logarithm (to which q-SDH 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:)
   $(g_1, A_0, A_1, \ldots, A_q)$, where $A_i = g_2^{x^i}$, for $i = 0, 1, \ldots, q$.
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 $c_{\text{type}} = 1$;)

(a) (Key setup)
$\mathcal{B}$ randomly selects y, z, r_i $(i = 1, \ldots, q-1)$ from $\mathbb{Z}_p^*$.
Let $f(X)$ be a polynomial of variable X such that $f(X) \leftarrow \prod_{i=1}^{q-1}(X + r_i) \bmod p = \sum_{i=0}^{q-1} \beta_i X^i$. (Here note that x is a value in $\mathbb{Z}_p^*$ determined by the input to $\mathcal{B}$ such as $A_1 = g_2^x$, while X is just a variable.) Clearly $\mathcal{B}$ 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

$$g_2' \leftarrow \prod_{i=0}^{q-1} A_i^{\beta_i} = g_2^{f(x)}, \qquad w_2' \leftarrow \prod_{i=0}^{q-1} A_{i+1}^{\beta_i} = g_2^{xf(x)} = (g_2')^x,$$
$$u_2' \leftarrow (g_2')^y, \qquad v_2' \leftarrow (g_2')^z.$$

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

$$\sigma_i \leftarrow \left(\prod_{j=0}^{q-2} \psi(A_j)^{\gamma_j}\right)^{m_i+y+s_i z} = \left(g_1^{f_i(x)}\right)^{m_i+y+s_i z} = \left(g_1^{f(x)/(x+r_i)}\right)^{m_i+y+s_i z}$$
$$= (g_1')^{(m_i+y+s_i z)/(x+r_i)} = \left((g_1')^{m_i}(u_1')(v_1')^{s_i}\right)^{1/(x+r_i)}.$$

$\mathcal{B}$ 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 $(g_1', g_2', w_2', u_2', v_2')$ and the distribution is exactly the same as that given by the signing oracle.

(c) (Output)
When $\mathcal{A}$ outputs a (valid) forgery $(m^*, \sigma^*, r^*, s^*)$, $\mathcal{B}$ checks whether $r^* \in \{r_1, \ldots, r_{q-1}\}$. If $r^* \in \{r_1, \ldots, r_{q-1}\}$, $\mathcal{B}$ then outputs `failure` and aborts. Otherwise, σ^* should satisfy

$$\sigma^* = (g_1')^{(m^*+y+s^* z)/(x+r^*)},$$

since $e(\sigma^*, w_2'(g_2')^{r^*}) = e(g_1', (g_2')^{m^*}(u_2')(v_2')^{s^*})$, $w' = (g_2')^x$, $u_2' = (g_2')^y, v_2' = (g_2')^z$ (i.e., $e(\sigma^*, (g_2')^{x+r^*}) = e(g_1', (g_2')^{m^*+y+s^* z}$.)
Let $c(X) \leftarrow \sum_{i=0}^{q-2} \omega_i X^i$ and $d \in \mathbb{Z}_p^*$ such that $f(X) \equiv c(X)(X + r^*) + d \pmod p$. $\mathcal{B}$ can efficiently calculate $\omega_i, d \in \mathbb{Z}_p^*$ $(i = 0, \ldots, q-2)$ from $f(X)$ and r^*.
$\mathcal{B}$ then computes

$$\eta \leftarrow \left((\sigma^*)^{1/(m^*+y+s^* z)} \prod_{i=0}^{q-2} \psi(A_i)^{-\omega_i}\right)^{1/d}$$
$$= \left((g_1^{f(x)})^{1/(x+r^*)} g_1^{-c(x)}\right)^{1/d} = \left(g_1^{(c(x)(x+r^*)+d)/(x+r^*)-c(x)}\right)^{1/d}$$
$$= \left(g_1^{c(x)+d/(x+r^*)-c(x)}\right)^{1/d} = g_1^{1/(x+r^*)}.$$

$\mathcal{B}$ outputs (η, r^*).

4. (If $c_{\mathrm{type}} = 2$;)
   (a) (Key setup)
   $\mathcal{B}$ randomly selects z, a, b, r_i $(i = 1, \ldots, q-1)$ from $\mathbb{Z}_p^*$, and randomly selects $k \in \{1, \ldots, q-1\}$. Let $f(X) \leftarrow \prod_{i=1}^{q-1}(X + r_i) \bmod p = \sum_{i=0}^{q-1} \beta_i X^i$, $f_i(X) \leftarrow f(X)/(X + r_i) \bmod p = \prod_{j=1, j\neq i}^{q-1}(X + r_j) \bmod p = \sum_{j=0}^{q-2} \gamma_j^{(i)} X^j$, $f_{k,i}(X) \leftarrow f(X)/((X + r_k)(X + r_i)) \bmod p = \prod_{j=1, j\neq i, j\neq k}^{q-1}(X + r_j) \bmod p = \sum_{l=0}^{q-3} \delta_l X^l$, $\mathcal{B}$ can efficiently calculate $\beta_i, \gamma_j, \delta_l \in \mathbb{Z}_p^*$ $(i = 0, \ldots, q-1, j = 0, \ldots, q-2, l = 0, \ldots, q-3)$.
   $\mathcal{B}$ computes

$$g_2' \leftarrow \prod_{i=0}^{q-2} A_i^{\gamma_i^{(k)}} = g_2^{f_k(x)}, \qquad w_2' \leftarrow \prod_{i=0}^{q-2} A_{i+1}^{\gamma_i^{(k)}} = g_2^{x f_k(x)} = (g_2')^x,$$

$$u_2' \leftarrow (\prod_{i=0}^{q-1} A_i^{\beta_i})^a (\prod_{i=0}^{q-2} A_i^{\gamma_i^{(k)}})^b = g_2^{af(x)} g_2^{-b f_k(x)}, \qquad v_2' \leftarrow (g_2')^z.$$

   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:
   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

$$\begin{aligned}\sigma_i \leftarrow\ & \Big(\prod_{j=0}^{q-2} \psi(A_j)^{\delta_j}\Big)^{m_i + s_i z} \Big(\prod_{j=0}^{q-2} \psi(A_j)^{\gamma_j^{(i)}}\Big)^a \Big(\prod_{j=0}^{q-2} \psi(A_j)^{\delta_j}\Big)^{-b} \\ &= (g_1^{f_{k,i}(x)})^{m_i + s_i z} (g_1^{a f_i(x)} g_1^{-b f_{k,i}(x)}), \\ &= (g_1')^{(m_i + y + s_i z)/(x + r_i)}.\end{aligned}$$

   $\mathcal{B}$ 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 $s_k \leftarrow (b - m_k)/z \bmod p$, and

$$\begin{aligned}\sigma_k \leftarrow\ & \Big(\prod_{i=0}^{q-2} \psi(A_i)^{\gamma_i^{(k)}}\Big)^a = g_1^{a f_k(x)} \\ &= (g_1^{f_k(x)(m_k + s_k z)} g_1^{a f(x)} g_2^{-b f_k(x)})^{1/(x + r_k)} \\ &= ((g_1')^{m_k} (v_1')^{s_k} u_1')^{1/(x + r_k)}.\end{aligned}$$

   Therefore, (σ_k, r_k, s_k) is a valid signature for public-key $(g_1', g_2', w_2', u_2', v_2')$.
   $\mathcal{B}$ 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 $r^* = r_k$ and $m^* + s^* z \not\equiv m_k + s_k z \pmod p$. If $r^* \neq r_k$ or $m^* + s^* z \equiv m_k + s_k z \pmod p$, $\mathcal{B}$ 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 $b = m_k + s_k z \bmod p$). Let $h(X) \leftarrow \sum_{i=0}^{q-3} \gamma_i X^i$ and $d \in \mathbb{Z}_p^*$ such that $f_k(X) \equiv h(X)(X + r_k) + e \pmod p$. Since $b^* \neq b$, $\mathcal{B}$ can compute

$$\eta \leftarrow \left(\frac{(\sigma^*/\sigma_k)^{1/(b^* - b)}}{\prod_{i=0}^{q-3} \psi(A_i)^{\gamma_i}}\right)^{1/e}.$$

Here,

$$\left(\frac{(\sigma^*/\sigma_k)^{1/(b^*-b)}}{\prod_{i=0}^{q-3}\psi(A_i)^{\gamma_i}}\right)^{1/e} = \left(\frac{g_1^{f_k(x)/(x+r_k)}}{g_1^{h(x)}}\right)^{1/e}$$
$$= g_1^{(f_k(x)/(x+r_k)-h(x))/e}$$
$$= g_1^{(e/(x+r_k)+h(x)-h(x))/e} = g_1^{1/(x+r_k)}$$

$\mathcal{B}$ outputs (η, r_k).

5. (If $c_{\text{type}} = 3$;)
   (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 ease of 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).
   Note that if the forgery $(m^*, \sigma^*, r^*, s^*)$ is Type-3, $m^* + s^* z' \equiv m_k + s_k z' \pmod p$ and $s^* \neq s_k$, so $z' = (m_k - m^*)/(s^* - s_k) \bmod p$.

This completes the description of algorithm $\mathcal{B}$. For any value of $c_{\text{type}} \in \{1, 2, 3\}$, the distribution of the simulation (public-key generation and signing oracle simulation) by $\mathcal{B}$ is exactly equivalent to that of the real attack scenario. Therefore, regardless of the value of c_{type} inside $\mathcal{B}$, adversary $\mathcal{A}$ produces a valid forgery in time t with probability at least ϵ.

We then obtain the probability ϵ' that $\mathcal{B}$ breaks the q-SDH assumption as follows:

- (When $c_{\text{type}} = 1$;)
  If Type-1 forgery occurs, $\mathcal{B}$ does not abort (i.e., breaks the q-SDH assumption).
- (When $c_{\text{type}} = 2$;)
  If Type-2 forgery occurs, $\mathcal{B}$ does not abort (i.e., breaks the q-SDH assumption) with probability $1/q_S$.
- (When $c_{\text{type}} = 3$;)
  If Type-3 forgery occurs, $\mathcal{B}$ does not abort (i.e., breaks the q-SDH assumption).

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

5 Variant of the Proposed Signature Scheme

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

5.1 Signature Scheme

The key generation is the same as that for the proposed signature scheme.

The signature, $(\sigma, \alpha, s, \mathtt{Test}(\alpha))$, of message, $m \in \mathbb{Z}_p^*$, is generated as follows:

$$\begin{aligned} &\sigma \leftarrow (g_1^m u_1 v_1^s)^{1/(\theta x+\rho)}, \quad \alpha \leftarrow g_2^{\theta x+\rho}, \quad \mathtt{Test}(\alpha) \leftarrow (U, V), \\ &U \leftarrow w_1^{1/\theta} g_1^{\lambda}, \quad V \leftarrow w_2^{\theta\lambda+\rho/\theta} g_2^{\rho\lambda}, \end{aligned} \tag{3}$$

where $\theta, \rho, s, \lambda \in \mathbb{Z}_p^*$ are randomly chosen.

The signature verification is:

$$\begin{aligned} &m, s \in \mathbb{Z}_p^*, \quad \sigma, U \in \mathbb{G}_1, \quad \alpha, V \in \mathbb{G}_2, \quad \sigma \neq 1, \quad \alpha \neq 1 \\ &e(\sigma, \alpha) = e(g_1, g_2^m u_2 v_2^s), \quad e(U, \alpha) = e(w_1, w_2) \cdot e(g_1, V). \end{aligned} \tag{4}$$

5.2 Security

The following theorem shows that this variant of the proposed signature scheme is existentially unforgeable against adaptive chosen message attacks, provided that the 2SDH assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$.

We now introduce a *strong* adaptive chosen message attack in this signature scheme, where, given adversary's query, message m, the signing oracle returns to the adversary the signature, (σ, r, s), of *the basic signature scheme* presented in Section 4, in place of the signature, $(\sigma, \alpha, s, \mathtt{Test}(\alpha))$, of *the variant signature scheme* introduced in this section.

If an adversary can forge the variant signature scheme against adaptive chosen message attacks (CMA), the adversary can also forge the variant signature scheme against *strong* adaptive chosen message attacks (S-CMA), since a reply from the signing oracle in S-CMA can be efficiently transformed into a reply from the signing oracle in CMA. That is, if the variant signature scheme is secure against S-CMA, the variant signature scheme is also secure against CMA. Thus, in the following theorem, we show that the variant signature scheme is secure against S-CMA.

Theorem 2. *If the (q_S, t', ϵ')-2SDH assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$, the proposed signature scheme is (t, q_S, ϵ)-existentially-unforgeable against "strong" adaptive chosen message attacks (S-CMA), 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 $\mathbb{G}_2$ and a single pairing operation.

Proof. Assume $\mathcal{A}$ is an adversary that (t, q_S, ϵ)-forges the signature scheme in the sense of "strong" adaptive chosen message attacks. We will then construct an algorithm $\mathcal{B}$ that breaks the q_S-2SDH assumption with (t', ϵ').

An informal outline of our proof is as follows: First we classify the output (forgery) of $\mathcal{A}$ into two types (Types-1,2). We will then show that any type of output allows $\mathcal{B}$ to break the 2SDH assumption. Type-1 forgery leads to breaking the 2SDH assumption. Type-2 forgery leads to breaking the discrete logarithm (to which 2SDH is reducible).

First, we introduce two 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 two types of forgers are as follows:

Type-1 forger outputs forged signature $(m^*, \sigma^*, \alpha^*, s^*)$ such that $m^* + s^* z \not\equiv m_i + s_i z \pmod{p}$ for all $i \in \{1, \ldots, q_S\}$.

Type-2 forger outputs forged signature $(m*, \sigma^*, \alpha^*, s^*)$ such that $m^* + s^* z \equiv m_k + s_k z \pmod{p}$ for some $k \in \{1, \ldots, q_S\}$. In this case, $s^* \neq s_k$, since $s^* = s_k$ implies $m^* = m_k$ and $m^* \neq m_k$. (Note that we only consider standard unforgeability, not strong unforgeability, here.)

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

1. (Input:)

   $(g_1, g_2, A, B, C_1, \ldots, C_{q_S}, a_1, \ldots, a_{q_S}, b_1, \ldots, b_{q_S})$, where $A = g_2^x$, $B = g_2^y$, $C_i = g_2^{\frac{y+b_i}{x+a_i}}$ $(i = 1, \ldots, q_S)$.
2. (Coin flip:)
   Algorithm $\mathcal{B}$ first picks random value $c_{\text{type}} \in \{1, 2\}$ that indicates its guess for the type of forger that $\mathcal{A}$ will emulate. The subsequent actions performed by $\mathcal{B}$ differ with $c_{\text{type}} \in \{1, 2\}$ as follows:
3. (If $c_{\text{type}} = 1$;)
   (a) (Key setup)
   $\mathcal{B}$ randomly selects $z \in \mathbb{Z}_p^*$, and $\mathcal{B}$ sets

   $$w_2 \leftarrow A = g_2^x, \qquad u_2 \leftarrow B = g_2^y, \qquad v_2 \leftarrow g_2^z.$$

   $\mathcal{B}$ 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:
   For each query $i \in \{1, 2, \ldots, q_S\}$ with message m_i from $\mathcal{A}$ to the signing oracle, $\mathcal{B}$ computes

   $$s_i \leftarrow (b_i - m_i)/z \bmod p, \qquad r_i \leftarrow a_i$$
   $$\sigma_i \leftarrow \psi(C_i) = g_1^{(y+b_i)/(x+a_i)} = g_1^{(m_i+y+s_i z)/(x+r_i)}.$$

   $\mathcal{B}$ returns (σ_i, r_i, s_i) to $\mathcal{A}$ as the reply to the query. Clearly this is a valid signature of the basic signature scheme for public-key $(g_1, g_2, w_2, u_2, v_2)$.
   (c) (Output) When $\mathcal{A}$ outputs a (valid) forgery $(m^*, \sigma^*, \alpha^*, s^*, \texttt{Test}(\alpha))$, $\mathcal{B}$ checks whether $m^* + s^* z \not\equiv m_i + s_i z \pmod{p}$ for all $i \in \{1, \ldots, q_S\}$. (Here $b_i = m_i + s_i z \pmod{p}$ for $i \in \{1, \ldots, q_S\}$.) If $m^* + s^* z \equiv m_k + s_k z \pmod{p}$ for some $k \in \{1, \ldots, q_S\}$, $\mathcal{B}$ outputs `failure` and aborts. Otherwise, $\mathcal{B}$ sets $b^* \leftarrow m^* + s^* z \bmod p$, and outputs $(\sigma^*, \alpha^*, b^*, \texttt{Test}(\alpha))$.
4. (If $c_{\text{type}} = 2$;)
   (a) (Key setup)
   $\mathcal{B}$ randomly selects x', y' from $\mathbb{Z}_p^*$.
   $\mathcal{B}$ computes

   $$w_2 \leftarrow g_2^{x'}, \qquad u_2 \leftarrow g_2^{y'}, \qquad v_2 \leftarrow A = g_2^x.$$

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

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

   $\mathcal{B}$ 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 $(m^*, \sigma^*, \alpha^*, s^*)$, $\mathcal{B}$ computes

$$z^* \leftarrow (m_i - m^*)/(s^* - s_i) \bmod p,$$

for all $i\{1, \ldots, q_S\}$ such that $s_i \neq s^*$, and checks whether $A = g_2^{z^*}$. If it holds, $z^* = z' = x$. $\mathcal{B}$ then randomly selects $c, d \in \mathbb{Z}_p^*$ and computes $g_1^{(y+d)/(x+c)}$ and g_2^c. $\mathcal{B}$ outputs $(g_1^{(y+d)/(x+c)}, g_2^{x+c}, d, \mathtt{Test}(\alpha))$.
Note that if forgery $(m^*, \sigma^*, \alpha^*, s^*)$ is Type-2, $m^* + s^* z' \equiv m_k + s_k z' \pmod p$ and $s^* \neq s_k$ for some $k \in \{1, \ldots, q_S\}$. Then, $z' = (m_k - m^*)/(s^* - s_k) \bmod p$.

This completes the description of algorithm $\mathcal{B}$. For any value of $c_{\mathrm{type}} \in \{1, 2\}$, the distribution of the simulation (public-key generation and signing oracle simulation) by $\mathcal{B}$ is exactly equivalent to that of the real attack scenario. Therefore, independently of the value of c_{type} inside $\mathcal{B}$, adversary $\mathcal{A}$ produces a valid forgery in time t with probability at least ϵ.

We then obtain the probability ϵ' that $\mathcal{B}$ breaks the 2SDH assumption as follows:

- (When $c_{\mathrm{type}} = 1$;)
  If Type-1 forgery occurs, $\mathcal{B}$ does not abort (i.e., breaks the 2SDH assumption).
- (When $c_{\mathrm{type}} = 2$;)
  If Type-2 forgery occurs, $\mathcal{B}$ does not abort (i.e., breaks the 2SDH assumption).

Since the value of c_{type} is independent of the type of forgery, $\mathcal{B}$ breaks the q_S-2SDH assumption with probability at least ϵ. ⊣

6 The Proposed Blind Signature Scheme

This section shows the application of the proposed signature scheme to blind signatures. We present a secure blind signature scheme in the standard model under the 2SDH and 2SDH assumptions.

6.1 Blind Signature Scheme

Let $(\mathbb{G}_1, \mathbb{G}_2)$ be bilinear groups as shown in Section 2.3. Here, we also assume that the message, m, to be blindly 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 $H : \{0,1\}^* \to \mathbb{Z}_p^*$, as mentioned in Section 3.5 in [7].

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

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 with the following signature generation protocol.
2. U randomly selects $s, t \in \mathbb{Z}_p^*$, computes

$$X \leftarrow g_1^{mt} u_1^t v_1^{st},$$

and sends X to S. Here $m \in \mathbb{Z}_p^*$ 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 using the witness indistinguishable proof as follows:

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

$$W \leftarrow g_1^{a_1} u_1^{a_2} v_1^{a_3},$$

and sends W to S.

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

(c) U computes

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

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

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

$$g_1^{b_1} u_1^{b_2} v_1^{b_3} = W X^{\eta}. \tag{5}$$

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 $x + r \equiv 0 \bmod p$, S tries again with a different random r. S also randomly selects $\ell \in \mathbb{Z}_p^*$, computes

$$Y \leftarrow (X v_1^{\ell})^{1/(x+r)},$$

and sends (Y, r, ℓ) to U.

Here, $Y = (X v_1^{\ell})^{1/(x+r)} = (g_1^m u_1 v_1^{s+\ell/t})^{t/(x+r)}$.

4. U randomly selects $f, \lambda \in \mathbb{Z}_p^*$, and computes

$$\tau \leftarrow (ft)^{-1} \bmod p, \qquad \sigma \leftarrow Y^{\tau}, \qquad \alpha \leftarrow w_2^f g_2^{fr}, \qquad \beta \leftarrow s + \ell/t \bmod p.$$

Compute $\texttt{Test}(\alpha) \leftarrow (U, V)$ as follows:

$$U \leftarrow w_1^{1/f} g_1^{\lambda}, \quad V \leftarrow w_2^{f\lambda + r} g_2^{fr\lambda}, \tag{6}$$

Here, $\sigma = (g_1^m u_1 v_1^{s+\ell/t})^{1/(fx+fr)} = (g_1^m u_1 v_1^{\beta})^{1/(fx+fr)}$, and $\alpha = w_2^f g_2^{fr} = g_2^{fx+fr}$.

5. $(\sigma, \alpha, \beta, \texttt{Test}(\alpha))$ is a blind signature of m.

Signature verification: Given public-key $(g_1, g_2, w_2, u_2, v_2)$, message m, and signature $(\sigma, \alpha, \beta, \texttt{Test}(\alpha))$, check

$$\begin{aligned} &m, \beta \in \mathbb{Z}_p^*, \quad \sigma, U \in \mathbb{G}_1, \quad \alpha, V \in \mathbb{G}_2, \quad \sigma \neq 1, \quad \alpha \neq 1, \\ &e(\sigma, \alpha) = e(g_1, g_2^m u_2 v_2^{\beta}), \quad e(U, \alpha) = e(w_1, w_2) \cdot e(g_1, V). \end{aligned} \tag{7}$$

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

6.2 Security

The following theorems show that the proposed blind signature scheme is perfectly blind and unforgeable provided that the 2SDH assumption hold in $(\mathbb{G}_1, \mathbb{G}_2)$.

Theorem 3. *The proposed blind signature scheme is perfectly blind.*

Proof. We will show that $\mathcal{A}$'s view is perfectly independent from the value of b in the blindness definition (experiment) shown in Section 2.2.

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 $\delta = (x + r)f \bmod p$ and $\beta \leftarrow s + \ell/t \bmod p$. Here, $\sigma = (g_1^m u_1 v_1^\beta)^{1/\delta}$, $\alpha = g_2^\delta$, 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). In addition, (α, U, V) is perfectly independent from (f, fr) (i.e., r).

That is, the distribution of the view of $\mathcal{S}^*$ for $\mathcal{U}_0$ and $\mathcal{U}_1$ in the blindness definition in Section 2.2 are equivalent.

In the blindness definition (experiment), whether $\mathcal{S}^*$ finally receives $\perp$ or two valid signatures depends only on whether $\mathcal{S}^*$'s reply (Y, R, ℓ) to $\mathcal{B}$ satisfies $e(Y, w_2 R) = e(X_i v_1^\ell, g_2)$ $(i = 0, 1)$ or not, and is independent from b, since the distributions of X_0 and X_1 are equivalent and the distribution of (X_0, X_1) is independent from b. ⊣

Definition 8. *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 assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$, the proposed blind signature scheme is (t, q_S, ϵ)-unforgeable against an L-interval synchronized run of adversaries, provided that*

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

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

Proof. Assume $\mathcal{A}$ is an adversary that (t, q_S, ϵ)-forges the blind signature scheme. We will then construct algorithm $\mathcal{B}$ that (t'', q_S, ϵ'')-forges the proposed signature scheme (the variant signature scheme: Section 5) with *strong* chosen message attacks (S-CMA). This leads to an algorithm that breaks the 2SDH 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 (Section 4) q_S times. Here note that we assume S-CMA, where the signing oracle return a signature with the form of (σ, r, β). 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

$$g_1^{b_1} u_1^{b_2} v_1^{b_3} = W X^{\eta}, \quad g_1^{b_1'} u_1^{b_2'} v_1^{b_3'} = W X^{\eta'}, \tag{8}$$

$\mathcal{B}$ can compute

$$\begin{aligned} m' &\leftarrow (b_1 - b_1')/(\eta - \eta') \bmod p, \\ t &\leftarrow (b_2 - b_2')/(\eta - \eta') \bmod p, \\ s' &\leftarrow (b_3 - b_3')/(\eta - \eta') \bmod p, \end{aligned} \tag{9}$$

such that

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

$\mathcal{B}$ computes

$$m \leftarrow m'/t \bmod p, \quad s \leftarrow s'/t \bmod p. \tag{10}$$

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

$$Y \leftarrow \sigma^t, \quad \ell \leftarrow t(\beta - s) \bmod p, \tag{11}$$

and returns $\mathcal{A}$ (Y, r, 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 the 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 $\mathcal{A}$.

Suppose that the synchronized run of blind signature generation protocol (including the WI protocol) consists of L sequential intervals, the j-th of which consists of the parallel execution of q_j rounds of the protocol, where $q_S = q_1 + \cdots + q_L$.

$\mathcal{B}$ then behaves in a synchronized run of $\mathcal{A}$ as follows:

1. The randomness of $\mathcal{G}$, $\mathcal{A}$ and $\mathcal{B}$ is randomly fixed. Then, q_S random challenges of $\mathcal{B}$, $(\eta_1, \ldots, \eta_{q_S})$, are also fixed. Hereafter in this procedure, the randomness except $\mathcal{B}$'s random challenges is fixed.
2. In the j-th interval of the synchronized run $(j = 1, \ldots, L)$, using $(\eta_{Q_j+1}, \ldots, \eta_{Q_j+q_j})$ $(Q_j = q_1 + \cdots q_{j-1}$ and $Q_1 = 0)$, $\mathcal{B}$ runs the blind signature generation protocol, and obtains $\mathcal{A}$'s responses, $(b_1, b_2, b_3)_{Q_j+1}$, $\ldots$, $(b_1, b_2, b_3)_{Q_j+q_j}$, for $\mathcal{B}$'s challenges, $(\eta_{Q_j+1}, \ldots, \eta_{Q_j+q_j})$.

3. $\mathcal{B}$ checks the validity. If all responses are valid, $\mathcal{B}$ rewinds the protocol up to the beginning of the j-th interval (i.e., just after $\mathcal{A}$'s sending the commitments, $(X,W)_{Q_j+1}, \ldots, (X,W)_{Q_j+q_j}$, to signer, $\mathcal{B}$). Otherwise, $\mathcal{B}$ halts.
   $\mathcal{B}$ then randomly selects challenges, $(\eta'_{Q_j+1}, \ldots, \eta'_{Q_j+q_j})$ $(\eta'_{Q_j+1} \neq \eta_{Q_j+1}, \ldots, \eta'_{Q_j+q_j} \neq \eta_{Q_j+q_j})$, and sends them to $\mathcal{A}$. $\mathcal{B}$ obtains $\mathcal{A}$'s responses, $(b_1,b_2,b_3)'_{Q_j+1}, \ldots, (b_1,b_2,b_3)'_{Q_j+q_j}$. $\mathcal{B}$ checks the validity, and if all responses are valid, computes $(m,t,s)_{Q_j+1}, \ldots, (m,t,s)_{Q_j+q_j}$ by Eqs. (8), (9) and (10). Otherwise, $\mathcal{B}$ return to the beginning of this step.
4. With the help of the signing oracle, $\mathcal{B}$ then computes $(Y,\ell)_{Q_j+1}, \ldots, (Y,\ell)_{Q_j+q_j}$ by Eq. (11), and sends them to $\mathcal{A}$.
5. Move to the next interval.
6. If all intervals are completed successfully, $\mathcal{A}$ provides an output.

We now consider the game scenario used in Definition 5. In the game scenario, let assume that the randomness of $\mathcal{G}, \mathcal{U}^*$ (here $\mathcal{A}$) and $\mathcal{S}$ except $\mathcal{S}$'s random challenges, $(\eta_1, \ldots, \eta_{q_S})$, is fixed. Thus value of $(\eta_1, \ldots, \eta_{q_S})$ chosen leads to the result of the game, $\mathcal{U}^*$'s win or not.

To characterize all choices of the value of $(\eta_1, \ldots, \eta_{q_S})$ that lead to win or not, we consider an L layer tree from its root node (the 0-th layer), in which all nodes in the $(j-1)$-th layer have $n_j \leftarrow (p-1)^{q_j}$ sons $(j = 1, \ldots, L)$. Thus, in total there are $n_1 n_2 \cdots n_L$ leafs at the bottom (the L-th layer). A path is labelled by $(p_1, p_2, \ldots, p_L)$, where $p_j \in \{1, \ldots, n_j\}$ $(j = 1, \ldots, L)$, identifies a j-th layer node on the path. So, each node can be identified by the corresponding path, i.e., $(p_1, p_2, \ldots, p_j)$ denotes the j-th layer node on path $(p_1, p_2, \ldots, p_L)$, and a leaf node (the L-th layer node) has the same label as the corresponding path. The root node is identified by $\perp$. Each path leads to the game result, 'win' or not ('lose'), and the leaf is labelled by 'win' or 'lose'. If a node leads to game abort, then the node is labelled by 'lose'. We say that a node survives if it is not labelled as 'lose'.

Let $\epsilon_{j-1,(p_1,\ldots,p_{j-1})}$ be the ratio of sons of $(j-1)$-th node $(p_1, \ldots, p_{j-1})$ that survive in the j-th layer among the n_j sons of node $(p_1, \ldots, p_{j-1})$. Let ϵ^* be $\mathtt{Adv}^{\mathrm{unforge}}_{\mathrm{PBS}}$ with the fixed randomness of $\mathcal{G}, \mathcal{U}^*$ $(\mathcal{A})$ and $\mathcal{S}$ except $\mathcal{S}$'s random challenges, $(\eta_1, \ldots, \eta_{q_S})$.

Therefore,

$$\epsilon^* = \sum_{(p_1,p_2,\ldots,p_{L-1})} (\epsilon_{0,\perp}\epsilon_{1,p_1}\epsilon_{2,(p_1,p_2)} \cdots \epsilon_{L-1,(p_1,p_2,\ldots,p_{L-1})})/(n_1 n_2 \cdots n_{L-1}),$$

where if $L = 1$, $\epsilon^* = \epsilon_{0,\perp}$.

We call path $(p_1, p_2, \ldots, p_L)$ *heavy* if $(\epsilon_{0,\perp} \cdots \epsilon_{L-1,(p_1,p_2,\ldots,p_{L-1})}) \geq \epsilon^*/2$. This leads to the following claim:

Claim. At least half of all 'win' paths are heavy.

Proof. Let's assume that more than half of all 'win' paths are not heavy. That is, more than $\epsilon^*(n_1 n_2 \cdots n_L)/2$ 'win' paths are not heavy. Each 'lose' path can be corresponded to a 'win' path disjointly (in the manner shown below), and there are $(1/(\epsilon_{0,\perp} \cdots \epsilon_{L-1,(p_1,p_2,\ldots,p_{L-1})}) - 1)$ 'lose' paths that correspond to a single 'win' path $(p_1, p_2, \ldots, p_L)$ disjointly.

This correspondence is made as follows: $(1 - \epsilon_{j-1,(p_1,p_2,\ldots,p_{j-1})})n_j$ 'lose' sons of $(j-1)$-th node $(p_1, \ldots, p_{j-1})$ and all their descendant paths can be corresponded to the $\epsilon_{j-1,(p_1,p_2,\ldots,p_{j-1})}n_j$ 'win' sons of node $(p_1, \ldots, p_{j-1})$. That is, $(1/\epsilon_{j-1,(p_1,p_2,\ldots,p_{j-1})} - 1)$ 'lose' sons of $(j-1)$-th node $(p_1, \ldots, p_{j-1})$ and their all descendant paths can be disjointly corresponded to each 'win' son of node $(p_1, \ldots, p_{j-1})$. By repeating this correspondence for all layers $(j = 1, \ldots, L)$, $(1/(\epsilon_{0,\perp} \cdots \epsilon_{L-1,(p_1,p_2,\ldots,p_{L-1})}) - 1)$ 'lose' paths are disjointly corresponded to each 'win' path $(p_1, p_2, \ldots, p_L)$.)

Therefore, if a 'win' path is not heavy, there are more than $2/\epsilon^*$ ('lose' or 'win') paths that correspond to the path. Since there are more than $\epsilon^*(n_1 n_2 \cdots n_L)/2$ 'not-heavy win' paths from the assumption, there must be more than $n_1 n_2 \cdots n_L$ $(= (2/\epsilon^*)(\epsilon^*(n_1 n_2 \cdots n_L)/2))$ paths. This contradicts that the total number of paths is $n_1 n_2 \cdots n_L$. ⊣

When path $(p_1, p_2, \ldots, p_L)$ is heavy, $\epsilon_{j-1,(p_1,p_2,\cdots,p_{j-1})} \geq \epsilon^*/2$ for all $j = 1, \ldots, L$. Therefore, if $\mathcal{B}$ tries $12 \ln(L+1)/\epsilon^*$ rewinding challenges in the j-th interval, the probability that $\mathcal{B}$ can compute $(m, t, s)_{Q_j+1}, \ldots, (m, t, s)_{Q_j+q_j}$ is at least $(1 - (L+1)^{-2})$ (by Chernoff bound). (Note that more precisely we have to consider the probability that $\eta'_{Q_j+i} = \eta_{Q_j+i}$ for some $i \in \{1, \ldots, q_j\}$. However, since the probability is negligible in n, around $q_j/(p-1)$, we ignore it in this evaluation (i.e., precisely we have to use $\epsilon^*/2 - q_j/(p-1)$ in place of $\epsilon^*/2$, but the difference is negligible in n).)

To summarise in total, $\mathcal{B}$ breaks the q_S-unforgeability of the basic signature scheme with probability at least $(1 - (L+1)^{-2})^L \epsilon^*/2$ (i.e., at least $(1 - (L+1)^{-1})\epsilon^*/2$) under the condition that $\mathcal{B}$ rewinds random challenges at most $12L \ln(L+1)/\epsilon^*$ times in L intervals.

The above-mentioned evaluation is based on ϵ^* as $\mathtt{Adv}_{\mathrm{PBS}}^{\mathrm{unforge}}$ under the fixed randomness of $\mathcal{G}, \mathcal{U}^*$ (here $\mathcal{A}$) and $\mathcal{S}$ except $\mathcal{S}$'s random challenges. Finally, given ϵ as $\mathtt{Adv}_{\mathrm{PBS}}^{\mathrm{unforge}}$ over the whole random space, we evaluate the success probability. Let $\mathcal{R}_0$ be the fixed part of randomness and $\mathcal{R}_1$ be $\mathcal{S}$'s randomness for challenges, $(\eta_1, \ldots, \eta_{q_S})$. We say that a value of $\mathcal{R}_0$ is heavy, if $\mathtt{Adv}_{\mathrm{PBS}}^{\mathrm{unforge}}$ under the value of $\mathcal{R}_0$ is at least $\epsilon/2$. This yields the following claim by a simple counting argument:

Claim. More than half of 'win' values of $(\mathcal{R}_0, \mathcal{R}_1)$ have heavy values of $\mathcal{R}_0$.

Thus, $\mathcal{B}$ breaks the q_S-unforgeability of the basic signature scheme with probability at least $(1 - 1/(L+1))\epsilon/8$ under the condition that $\mathcal{B}$ rewinds random challenges at most $24L \ln(L+1)/\epsilon$ times in total. By combining this result with Theorem 2 we obtain the claim of this theorem. ⊣

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. A synchronized run is 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.

7 The Proposed Partially Blind Signature Scheme

This section shows the application of the proposed signature scheme to partially blind signatures. We present a secure partially blind signature scheme in the standard model under the 2SDH assumption.

7.1 Partially Blind Signature Scheme

Let $(\mathbb{G}_1, \mathbb{G}_2)$ be bilinear groups as shown in Section 2.3. Here, we also assume that the message, m, to be partially blindly 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 $H : \{0,1\}^* \to \mathbb{Z}_p^*$, as mentioned in Section 3.5 in [7].

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 2.2) in a 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 $x + r \equiv 0 \bmod p$, S tries again with a different random r. S also randomly selects $\ell \in \mathbb{Z}_p^*$, computes

$$Y \leftarrow (X v_1^{\ell})^{1/(x+r)},$$

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

$$\tau = (ft)^{-1} \bmod p, \qquad \sigma \leftarrow Y^{\tau}, \qquad \alpha \leftarrow w_2^f g_2^{fr}, \qquad \beta \leftarrow s + \ell/t \bmod p.$$

Compute $\mathtt{Test}(\alpha) \leftarrow (U, V)$ as follows:

$$U \leftarrow w_1^{1/f} g_1^{\lambda}, \quad V \leftarrow w_2^{f\lambda + r} g_2^{fr\lambda}, \tag{12}$$

Here, $\sigma = (h_1^{m_0} g_1^{m_1} u_1 v_1^{s+\ell/t})^{1/(fx+fr)} = (h_1^{m_0} g_1^{m_1} u_1 v_1^{\beta})^{1/(fx+fr)}$, and $\alpha = w_2^f g_2^{fr} = g_2^{fx+fr}$.

5. $(\sigma, \alpha, \beta, \mathtt{Test}(\alpha))$ 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 a public-key $(g_1, g_2, w_2, u_2, v_2, h_2)$, common information m_0, message m_1, and a signature $(\sigma, \alpha, \beta, \texttt{Test}(\alpha))$, check

$$m_0, m_1 \in \mathbb{Z}_p^*, \quad \beta \in \mathbb{Z}_p \quad \sigma, U \in \mathbb{G}_1, \quad \alpha, V \in \mathbb{G}_2, \quad \sigma \neq 1, \quad \alpha \neq 1,$$
$$e(\sigma, \alpha) = e(g_1, h_2^{m_0} g_2^{m_1} u_2 v_2^{\beta}), \qquad e(U, \alpha) = e(w_1, w_2) \cdot e(g_1, V).$$

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

7.2 Performance

The performance of these partially blind signatures is almost the same as that of the proposed blind signatures.

7.3 Security

The following theorems show that the proposed partially blind signature scheme is unforgeable and perfectly blind provided that the 2SDH assumption hold in $(\mathbb{G}_1, \mathbb{G}_2)$.

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 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*

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

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

Proof. Assume $\mathcal{A}$ is an adversary that (t, q_S, ϵ)-forges the partially blind signature scheme. We will then construct algorithm $\mathcal{B}$ that $(t + \Theta(qT), q_S, \epsilon/2)$-forges our proposed blind signature scheme. This leads to an algorithm that breaks the q-2SDH assumption with $(q_S, O\Big(\frac{24L \ln (L+1)}{\epsilon} \cdot t\Big) + \Theta(q_S T), \frac{1-1/(L+1)}{16} \cdot \epsilon)$ by Theorem 4.

First, we introduce two types of forgers, $\mathcal{A}$. Let $(g_1, g_2, w_2, u_2, v_2, h_2)$ be given to $\mathcal{A}$ as a public-key, and $\xi \leftarrow \log_{g_2} h \in \mathbb{Z}_p^*$ (i.e., $h = g_2^{\xi}$). Then there are two types of forgers, $\mathcal{A}$, that output at least $q_S + 1$ signatures $(m_{0,1}, m_{1,1}, \sigma_1, \alpha_1, \beta_1), \ldots, (m_{0,L}, \sigma_L, \alpha_L, \beta_L)$ $(L \geq q_S + 1)$ as follows:

Type-1 forger $(\xi m_{0,i} + m_{1,i} \bmod p, \sigma_i, \alpha_i, \beta_i) \neq (\xi m_{0,j} + m_{1,j} \bmod p, \sigma_j, \alpha_j, \beta_j)$ for all pairs (i, j) such that $i \neq j$ and $i, j \in \{1, \ldots, L\}$.

Type-2 forger There exists pair (i, j) such that $i \neq j$, $i, j \in \{1, \ldots, L\}$ and $(\xi m_{0,i} + m_{1,i} \bmod p, \sigma_i, \alpha_i, \beta_i) = (\xi m_{0,j} + m_{1,j} \bmod p, \sigma_j, \alpha_j, \beta_j)$.

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

1. (Input:)
   (g_1, g_2, w, u, v) as a public key of the blind signature scheme.

2. (Coin flip:)
   Algorithm $\mathcal{B}$ first picks a random value $c \in \{1,2\}$ that indicates its guess for the type of forger that $\mathcal{A}$ will emulate. The subsequent actions performed by $\mathcal{B}$ differ with $c_{\text{type}} \in \{1,2\}$ as follows:
3. (If $c = 1$;)
   (a) (Key setup)
   $\mathcal{B}$ randomly selects $\xi \in \mathbb{Z}_p^*$ and computes $h \leftarrow g_2^{\xi}$. $\mathcal{B}$ then provides (g_1, g_2, w, u, v, h) to $\mathcal{A}$ as a public key of the partially blind signatures.
   (b) (Simulation)
   In the partially blind signing process, $\mathcal{A}$ sends to $\mathcal{B}$ X along with common information m_0 and the witness indistinguishable proof of witness (m_1, s, R) for $Z = X/(\psi(h)^{m_0}\psi(u)) = g_1^{m_1}\psi(v)^s(\psi(wf))^R$. In the process, $\mathcal{B}$ sends X to S when $\mathcal{A}$ sends out X. $\mathcal{B}$ also transfers a_1, a_2, a_3 from $\mathcal{A}$ to S, and transfers η from S to $\mathcal{A}$. When $\mathcal{A}$ replies (b_1, b_2, b_3), $\mathcal{B}$ computes $b_1' \leftarrow b_1 + \eta\xi m_0 \bmod p$, and replies (b_1', b_2, b_3) to S. If $\mathcal{A}$'s proof is valid, the proof changed by $\mathcal{B}$ is also a valid proof of witness $(\xi m_0 + m_1 \bmod p, s, R)$ for $X/\psi(u)$.
   (c) (Output)

   It is easy for $\mathcal{B}$ to convert $\mathcal{A}$'s output (at least $q_S + 1$ signatures) to $\mathcal{B}$'s output (at least $q_S + 1$ signatures) such that $\mathcal{A}$'s signature (σ, α, β) for (m_0, m_1) is $\mathcal{B}$'s signature (σ, α, β) for $m \leftarrow \xi m_0 + m_1 \bmod p$.
   If $(\xi m_{0,i} + m_{1,i} \bmod p, \sigma_i, \alpha_i, \beta_i) \neq (\xi m_{0,j} + m_{1,j} \bmod p, \sigma_j, \alpha_j, \beta_j)$ for all pairs (i,j) such that $i \neq j$ and $i, j \in \{1, \ldots, L\}$, then $\mathcal{B}$ obtains L distinct message/signatures. Otherwise, $\mathcal{B}$ aborts.
4. (If $c = 2$;)
   (a) (Key setup)
   $\mathcal{B}$ randomly selects $x', z' \in \mathbb{Z}_p^*$, computes $w' \leftarrow g_2^{x'}$ and $v' \leftarrow g_2^{z'}$, and sets $h \leftarrow w$. $\mathcal{B}$ then provides (g_1, g_2, w', u, v', h) to $\mathcal{A}$ as a public key of the partially blind signatures.
   (b) (Simulation)
   $\mathcal{B}$ can play the role of a signer in the partially blind signing process with $\mathcal{A}$, since $\mathcal{B}$ knows secret key (x', z').
   (c) (Output)
   If there exists a pair (i,j) such that $i \neq j$, $i, j \in \{1, \ldots, L\}$ and $(xm_{0,i} + m_{1,i} \bmod p, \sigma_i, \alpha_i, \beta_i) = (xm_{0,j} + m_{1,j} \bmod p, \sigma_j, \alpha_j, \beta_j)$, where $h' = w = g_2^x$, then $m_{0,i} \neq m_{0,j}$ since $m_{0,i} = m_{0,j}$ implies $(m_{0,i}, m_{1,i}, \sigma_i, \alpha_i, \beta_i) = (m_{0,j}, m_{1,j}, \sigma_j, \alpha_j, \beta_j)$, which is not allowed. Therefore, $\mathcal{B}$ can calculate $x = (m_{1,j} - m_{1,i})/(m_{0,i} - m_{0,j}) \bmod p$. $\mathcal{B}$ then can forge L signatures in the public-key (g_1, g_2, w, u, v), since $\mathcal{B}$ knows the secret-key x.
   Otherwise, $\mathcal{B}$ aborts.

This completes the description of algorithm $\mathcal{B}$. In any value of $c \in \{1,2\}$, the distribution of the simulation (public-key generation and simulation) by $\mathcal{B}$ is exactly equivalent to that of the real attack scenario. Therefore, regardless of the value of c inside $\mathcal{B}$, adversary $\mathcal{A}$ produces a valid forgery in time t with probability at least ϵ.

We then obtain the probability ϵ'' that $\mathcal{B}$ forges our blind signatures as follows:

- (When $c = 1$;)
  If Type-1 forgery occurs, $\mathcal{B}$ does not abort (i.e., forges our blind signatures).
- (When $c = 2$;)
  If Type-2 forgery occurs, $\mathcal{B}$ does not abort (i.e., forges our blind signatures).

Since the value of c is independent from which type of forgery occurs, $\mathcal{B}$ forges our blind signatures with probability at least $1/2$. $\dashv$

7.4 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.

8 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 [25], and uses Paillier encryption for a simulator to extract blind messages with the help of the CRS model, and also uses a trapdoor commitment [17] 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.

8.1 Converted 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 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*, [17] are also generated. We consider that (c_0, c_1) is a part of the system parameters such as the specification of $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$.

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 with the following signature generation protocol.
2. U randomly selects $s, t \in \mathbb{Z}_p^*$ and $A \in \mathbb{Z}_{N^2}$, computes

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

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 equations hold 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 they hold, S accepts. Otherwise, S rejects and aborts.

(The remaining procedure is the same as that of the original blind signature scheme as follows:)

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

$$Y \leftarrow (Xv_1^{\ell})^{1/(x+r)},$$

and sends (Y, r, ℓ) to U.
4. U randomly selects $f, \lambda \in \mathbb{Z}_p^*$, and computes

$$\tau \leftarrow (ft)^{-1} \bmod p, \qquad \sigma \leftarrow Y^{\tau},$$
$$\alpha \leftarrow w_2^f g_2^{fr}, \qquad \beta \leftarrow s + \ell/t \bmod p.$$

Compute $\texttt{Test}(\alpha) \leftarrow (U, V)$ as follows:

$$U \leftarrow w_1^{1/f} g_1^{\lambda}, \quad V \leftarrow w_2^{f\lambda + r} g_2^{fr\lambda}, \tag{13}$$

5. $(\sigma, \alpha, \beta, \texttt{Test}(\alpha))$ is a blind signature of m.

Signature verification: (Same as that of the original blind signature scheme as follows:)
Given public-key $(g_1, g_2, w_2, u_2, v_2)$, message m, and signature $(\sigma, \alpha, \beta, \texttt{Test}(\alpha))$, check

$$m, \beta \in \mathbb{Z}_p^*, \quad \sigma, U \in \mathbb{G}_1, \quad \alpha, V \in \mathbb{G}_2, \quad \sigma \neq 1, \quad \alpha \neq 1,$$
$$e(\sigma, \alpha) = e(g_1, g_2^m u_2 v_2^{\beta}), \quad e(U, \alpha) = e(w_1, w_2) \cdot e(g_1, V). \tag{14}$$

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

8.2 Properties:

The signature verification of the converted blind signature scheme is equivalent to that of the original one. That is, verifiers do not need to care about whether a signature to be verified is the original or a converted one. In addition, the original signing protocol is employed by the converted one as a core part as it is. The public-key and secret-key can be also shared by the original and converted schemes.

Therefore, a signer can flexibly choose the original signing protocol or the converted one, depending on the signer's strategy, whether the signer permits a fully concurrent run of users or not. (The signer can control a run of users; e.g., a signer can reject replying to a user if the user's response is received too late after receiving the commitment, or control the timing at which the challenges are sent to users.) For example, a signer can change the choice of the signing protocol day by day or event by event.

A signer and users can flexibly select and change CRS without changing the public-key of the signer. Even if the underlying CRS is broken (e.g., the secret key is revealed and published), it does not affect the security of the signatures, since CRS is only employed in an online manner during the signing protocol, and the key of the signatures is independent from CRS.

8.3 Security:

The following theorems show that the proposed blind signature scheme is blind and unforgeable.

Informally, the signature generation protocol is (statistically) witness indistinguishable, WI, except D, which is Paillier encryption of a message. So, this blind signature scheme satisfies blindness under the Paillier encryption is semantically secure.

As for unforgeability, if the WI protocol in the signature generation protocol is accepted by a signer, the security reduction algorithm for unforgeability (or a simulator of the signer), who knows the trapdoor of CRS (i.e., P, Q), can extract (m, s, t) by decrypting ED^{η} (for two distinct values of η) without rewinding $\mathcal{A}$. So, the security reduction works against any concurrent run of adversaries.

Theorem 7. *The proposed blind signature scheme is blind assuming that Paillier encryption is semantically secure (IND-CPA).*

Proof. We assume that the proposed blind signature scheme is not blind, i.e., assume that a dishonest signer $\mathcal{S}^*$ can guess b correctly with a non-negligible (in n) advantage, ϵ, in the definition of blindness shown in Section 2.2. We can then construct an algorithm $\mathcal{B}$ to break the semantic security of Paillier encryption as follows:

1. Given public-key (N, G) of Paillier encryption, $\mathcal{B}$ generates the public and secret keys, (pk, sk), of a trapdoor commitment, *commit*, and gives $((N, G), pk)$ to $\mathcal{S}^*$ as CRS.
2. $\mathcal{S}^*$ provides $\mathcal{B}$ with a public-key, $(g_1, g_2, w_2, u_2, v_2)$, and two messages $m_0, m_1 \in \mathbb{Z}_p^*$. $\mathcal{S}$ checks that $g_1 \in \mathbb{G}_1$, $g_2, w_2, u_2, v_2 \in \mathbb{G}_2$, and $m_0, m_1 \in \mathbb{Z}_p^*$. If it does not hold, $\mathcal{B}$ halts and outputs $\beta' \in \{0, 1\}$ randomly.
3. $\mathcal{B}$ randomly selects $s_0, t_0, s_1, t_1 \in \mathbb{Z}_p^*$, and sets $M_0 \leftarrow (m_0 t_0 \mod p) + t_0 2^K + (s_0 t_0 \mod p) 2^{2K}$ and $M_1 \leftarrow (m_1 t_1 \mod p) + t_1 2^K + (s_1 t_1 \mod p) 2^{2K}$.
4. $\mathcal{B}$ gives (M_0, M_1) to the encryption oracle, $\mathcal{EO}$, of Paillier encryption. $\mathcal{EO}$ randomly selects $\beta \in \{0, 1\}$ and $A_0 \in \mathbb{Z}_{N^2}$, encrypts M_b to $D \leftarrow G^{M_\beta} A_0^N \bmod N^2$, and gives D to $\mathcal{B}$.
5. $\mathcal{B}$ randomly selects $b \in \{0, 1\}$ and $A_1 \in \mathbb{Z}_{N^2}$, and computes

$$X_0 \leftarrow g_1^{m_0 t_0} u_1^{t_0} v_1^{s_0 t_0}, \quad D_0 \leftarrow D,$$
$$X_1 \leftarrow g_1^{m_1 t_1} u_1^{t_1} v_1^{s_1 t_1}, \quad D_1 \leftarrow G^{(m_1 t_1 \mod p) + t_1 2^K + (s_1 t_1 \mod p) 2^{2K}} A_1^N \bmod N^2,$$

   and sends $\mathcal{S}^*$ (X_b, D_b) as $\mathcal{U}_0$'s request and $(X_{\bar{b}}, D_{\bar{b}})$ as $\mathcal{U}_1$'s request.
6. $\mathcal{B}$ executes the protocol to prove to $\mathcal{S}^*$ that $\mathcal{B}$ knows $(m_i t_i \bmod p, t_i, s_i t_i \bmod p)$ regarding (X_i, D_i) $(i = 0, 1)$ for $\mathcal{U}_0$ and $\mathcal{U}_1$. $\mathcal{B}$'s procedure regarding (X_1, D_1) for $\mathcal{U}_{\bar{b}}$ is exactly the same as the real one by $\mathcal{U}_{\bar{b}}$. $\mathcal{B}$'s procedure regarding X_0 for $\mathcal{U}_b$ is exactly the same as the

real one by $\mathcal{U}_b$, but then procedure regarding $D_0 (= D)$ for $\mathcal{U}_b$ is a fake procedure that uses the trapdoor, sk, of trapdoor commitment *commit*. That is, after obtaining η from $\mathcal{S}^*$, $\mathcal{B}$ randomly generates $F_0 \in \mathbb{Z}_{N^2}$ and computes $E_0 \leftarrow G^{b_1+b_2 2^K+b_3 2^{2K}} F_0^N / D_0^\eta \bmod N^2$, where (b_1, b_2, b_3) is determined by the above-mentioned procedure regarding X_0. $\mathcal{B}$ then opens C to (E_0, r^*) with $C = commit(E_0, r^*, pk)$ using trapdoor sk, and sends F_0 and (E_0, r^*) as well as (b_1, b_2, b_3) to $\mathcal{S}^*$.

7. After completing the protocol above, the remaining procedure is the same as the real one (i.e., $\mathcal{B}$ acts as a honest user).
8. After completing the signing phase of the proposed blind signature for $\mathcal{U}_0$ and $\mathcal{U}_1$, $\mathcal{B}$ checks the validity of the two obtained signatures for $\mathcal{U}_0$ and $\mathcal{U}_1$. If at least one is invalid, $\mathcal{B}$ gives $\perp$ to $\mathcal{S}^*$. If both of them are valid, $\mathcal{B}$ gives them to $\mathcal{S}^*$, $\mathcal{B}$ then obtains the output of $\mathcal{S}^*$, b'. If $b = b'$, $\mathcal{B}$ outputs $\beta' \leftarrow 0$, otherwise outputs $\beta' \leftarrow 1$.

If $\beta = 0$, the distributions of the views of $\mathcal{S}^*$ for $\mathcal{U}_0$ and $\mathcal{U}_1$ in the above-mentioned $\mathcal{B}$'s procedure are identical to those in the experiment of blindness of the proposed blind signature scheme (in Section 2.2).

Therefore, $\Pr[b = b' \mid \beta = 0] = 1/2 + \epsilon/2$.

If $\beta = 1$, the distributions of D_0 and D_1 are exactly the same. The distributions of the views of $\mathcal{S}^*$ on the protocols regarding X_0 and X_1 are statistically indistinguishable, and the distributions of the views of $\mathcal{S}^*$ on the fake protocol regarding D_0 and on the real protocol regarding D_1 are also statistically indistinguishable.

Whether $\mathcal{B}$ gives $\mathcal{S}^*$ $\perp$ or two valid signatures depends only on whether $\mathcal{S}^*$'s reply (Y, R, ℓ) to $\mathcal{B}$ satisfies $e(Y, w_2 R) = e(X_i v_1^\ell, g_2)$ $(i = 0, 1)$ or not, but does not depend on the value of b (or independent from b), since the distributions of X_0 and X_1 are equivalent.

Hence, $|\Pr[b \neq b' \mid \beta = 1] - 1/2| < \mu$, where μ is negligible in n.

Thus,

$$\begin{aligned}
\Pr[\beta = \beta'] &= \Pr[\beta = \beta' = 0 \vee \beta = \beta' = 1] \\
&= \Pr[\beta' = 0 \mid \beta = 0] \Pr[\beta = 0] + \Pr[\beta' = 1 \mid \beta = 1] \Pr[\beta = 1] \\
&= 1/2(\Pr[b = b' \mid \beta = 0] + \Pr[b \neq b' \mid \beta = 1]) \\
&> 1/2(1/2 + \epsilon/2 + 1/2 - \mu) = 1/2 + \epsilon/4 - \mu/2.
\end{aligned}$$

Therefore, the advantage of IND-CPA of $\mathcal{B}$ against Paillier encryption, $2 \cdot \Pr[\beta = \beta'] - 1$, is non-negligible, $(\epsilon/2 - \mu)$ in n, which contradicts the assumption. $\dashv$

Theorem 8. *Let the 2SDH assumption hold in $(\mathbb{G}_1, \mathbb{G}_2)$, and the underlying commitment, commit, satisfy the binding condition. Then, the proposed blind signature scheme is unforgeable against a concurrent run of adversaries in the CRS model.*

Proof. The top-level strategy of the proof is similar to that of Theorem 4. That is, first we assume $\mathcal{A}$ is an adversary that forges the blind signature scheme with non-negligible probability *epsilon* in n. We will then construct algorithm $\mathcal{B}$ that forges the proposed signature scheme (basic signature scheme presented in Section 5) with non-negligible probability in n. This leads to an algorithm that breaks the 2SDH assumption by Theorem 2. Here, in the reduction, we also assume that $\mathcal{A}$ can break the binding condition with negligible probability μ in n(due to the theorem statement).

The major difference of this proof from the proof of Theorem 4 is that, in this proof, $\mathcal{B}$ does not rewind $\mathcal{A}$, instead $\mathcal{B}$ simulates the rewinding by itself with help from the trapdoor of the commitment.

$\mathcal{B}$ behaves with a concurrent run of $\mathcal{A}$ as follows:

1. Given $(g_1, g_2, w_2, u_2, v_2)$ as a public key of the basic signature scheme, $\mathcal{B}$ randomly selects secret and public keys of Paillier encryption, P and Q, and $(N = PQ, G)$, and the public and secret keys, (pk, sk), of a trapdoor commitment, *commit*.
   $\mathcal{B}$ provides $\mathcal{A}$ $(g_1, g_2, w_2, u_2, v_2)$ as a public key for blind signatures as well as (N, G, pk) as CRS.
2. $\mathcal{B}$ executes the blind signing protocol with $\mathcal{A}$, where $\mathcal{B}$ plays the role of an honest signer against user $\mathcal{A}$ (dishonest user), who tries to forge a signature.
3. On receiving the j-th signing request message, (X, D) and (W, C), $\mathcal{B}$ returns a random challenge, $\eta \in \mathbb{Z}_p^*$, to $\mathcal{A}$. When $\mathcal{B}$ later (at some step) receives the corresponding response, $(b_1, b_2, b_3, F, E, r^*)$, from $\mathcal{A}$, $\mathcal{B}$ checks whether the following equations hold 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 the equations do not hold, $\mathcal{B}$ halts and outputs `failure`.
4. If the equations hold, $\mathcal{B}$ randomly selects $\eta' \in \mathbb{Z}_p^*$ and computes $\xi \in \mathbb{Z}_N$ such that $G^{\xi} U^N \equiv ED^{\eta'} \pmod{N^2}$ by using secret key, (P, Q), of Paillier encryption. If there exist (b_1', b_2', b_3') such that $\xi = b_1' + b_2' 2^K + b_3' 2^{2K}$, and $|b_i'| \leq (2 + c_1)|p| \quad (i = 1, 2, 3)$, then $\mathcal{B}$ checks whether the following equation holds or not: $g_1^{b_1'} u_1^{b_2'} v_1^{b_3'} = WX^{\eta'}$.
   If it does not hold, return to the top of this step. The number of iterations of this step is bounded by a polynomial in security parameter n.
   If it holds, $\mathcal{B}$ computes

$$t \leftarrow (b_2 - b_2')/(\eta - \eta') \bmod p, \quad m \leftarrow (b_1 - b_1')/(t(\eta - \eta')) \bmod p,$$

$$s \leftarrow (b_3 - b_3')/(t(\eta - \eta')) \bmod p,$$

   where $X = g_1^{mt} u_1^t v_1^{st}$.
5. Since $\mathcal{B}$ is allowed to access the signing oracle of the basic signature scheme, $\mathcal{B}$ sends m to the signing oracle. The signing oracle returns (σ, R, β) to $\mathcal{B}$ such that $\sigma \leftarrow (g_1^m u_1 v_1^{\beta})^{1/(x+r)}$.
   $\mathcal{B}$ computes

$$Y \leftarrow \sigma^t, \quad \ell \leftarrow t(\beta - s) \bmod p, \tag{15}$$

   and returns (Y, r, ℓ) to $\mathcal{A}$.
6. If the whole signing procedures with q_S rounds are completed successfully, $\mathcal{A}$ provides an output.
7. If, in the output of $\mathcal{A}$, $\mathcal{B}$ finds a valid signature of a message, m^*, that is different from the messages given to the signing oracle, $\mathcal{B}$ outputs the signature with m^*. Otherwise, $\mathcal{B}$ outputs `failure`.

After all q_S rounds of the above procedures are completed, successful adversary $\mathcal{A}$ outputs the 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.

We now analyze the success probability of $\mathcal{B}$.

We assume that $\mathcal{A}$ forges the blind signature scheme with non-negligible (in n) probability ϵ and breaks the binding condition of *commit* with negligible (in n) probability μ. (Here, to break the binding condition of *commit* means that $\mathcal{A}$ opens two distinct values for the same commitment at least once during its execution.)

We then consider the case, `Forge-Bind`, where $\mathcal{A}$ forges the blind signature scheme but does not break the binding condition of *commit*. `Forge-Bind` occurs with probability at least $\epsilon^* \leftarrow \epsilon - \mu$.

Similar to the analysis in the proof of Theorem 4, we consider the game scenario (in Definition 5) with concurrent runs of users.

We assume that the randomness of $\mathcal{G}$, $\mathcal{U}^*$ (here $\mathcal{A}$) and $\mathcal{S}$ except $\mathcal{S}$'s random challenges, $(\eta_1, \ldots, \eta_{q_S})$, is fixed. It follows that the choice of the value of $(\eta_1, \ldots, \eta_{q_S})$ leads to the result of the game, $\mathcal{U}^*$'s win (`Forge-Bind`) or not. The probability of `Forge-Bind`, ϵ^*, is taken over the whole random space. Let $\mathcal{R}_0$ be the fixed part of randomness and $\mathcal{R}_1$ be $\mathcal{S}$'s randomness for challenges, $(\eta_1, \ldots, \eta_{q_S})$. We say that a value of $\mathcal{R}_0$ is heavy, if the probability that `Forge-Bind` occurs under the value of $\mathcal{R}_0$ is at least $\epsilon^+ \leftarrow \epsilon^*/2$. So we have the following claim by a simple counting argument:

Claim. More than half of the values of $(\mathcal{R}_0, \mathcal{R}_1)$ that lead to case `Forge-Bind` have heavy values of $\mathcal{R}_0$.

Hereafter, we analyze the success probability of $\mathcal{B}$ based on the probability of `Forge-Bind` that is at least ϵ^+, under a fixed heavy value of $\mathcal{R}_0$,

To characterize all choices of the value of $(\eta_1, \ldots, \eta_{q_S})$ that lead to `Forge-Bind` or not, we consider a q_S layer tree from a root node (the 0-th layer), in which all nodes have $(p-1)$ sons; in total there are $(p-1)^{q_S}$ leafs at the bottom (the q_S-th layer). A path is labelled by $(\eta_1, \eta_2, \ldots, \eta_{q_S})$, which identifies a j-th layer node on the path $(j = 1, \ldots, q_S)$. That is, each node can be identified by the corresponding path, i.e., $(\eta_1, \eta_2, \ldots, \eta_j)$ denotes the j-th layer node on path $(\eta_1, \eta_2, \ldots, \eta_{q_S})$, and a leaf node has the same label as the corresponding path. The root node is identified by $\perp$. Each path leads to the game result, 'win' (`Forge-Bind`) or not ('lose'), and the leaf is labelled by 'win' or 'lose'.

Each node $(\eta_1, \ldots, \eta_{j-1})$ $(j = 1, \ldots, q_S)$ (if $j = 1$, node $(\eta_1, \ldots, \eta_{j-1})$ denotes the root node, $\perp$) corresponds to the j-th commitment value, $\mathtt{com}_{(\eta_1, \ldots, \eta_{j-1})}$, of the user after signer's $(j-1)$ challenges $(\eta_1, \ldots, \eta_{j-1})$. In a concurrent run of our protocol, after the user sends j-th commitment value $\mathtt{com}_{(\eta_1, \ldots, \eta_{j-1})}$ $(j = 1, \ldots, q_S)$, the signer sends a challenge, η_j, and then later the user sends response $\mathtt{resp}_j$ corresponding to $\mathtt{com}_{(\eta_1, \ldots, \eta_{j-1})}$ and η_j. Here, $\mathtt{com}_{(\eta_1, \ldots, \eta_{j-1})}$ corresponds to (X, D, W, C) and $\mathtt{resp}_j$ to $(b_1, b_2, b_3, F, E, r^*)$, in our protocol description.

The timing and validity of user's response $\mathtt{resp}_j$ depend on the user's strategy and signer's challenges $(\eta_1, \ldots, \eta_l)$ that are sent before the user sends $\mathtt{resp}_j$. If $\mathtt{resp}_j$ is accepted (or rejected) as a valid response to $\mathtt{com}_{(\eta_1, \ldots, \eta_{j-1})}$ and η_j, then $b_j \leftarrow 1$ (or $b_j \leftarrow 0$) is labelled as η_j. When a path $(\eta_1, \eta_2, \ldots, \eta_{q_S})$ is fixed, then the value of b_j labelled to η_j is fixed for all $j = 1, \ldots, q_S$. Note that η_j along with $(\eta_1, \eta_2, \ldots, \eta_{j-1})$ cannot be always labelled by a unique value of b_j without the suffix of the path, $(\eta_{j+1}, \eta_{j+2}, \ldots)$.

We now analyze the success probability of $\mathcal{B}$ based on this tree model.

As shown in the algorithm of $\mathcal{B}$, after receiving the j-th correct response $(b_1, b_2, b_3, F, E, r^*)$ to the j-th challenge η_j and $\mathtt{com}_{(\eta_1, \ldots, \eta_{j-1})}$ and η_j from $\mathcal{A}$, $\mathcal{B}$ can calculate the possible response to any other j-th challenge η_j', and can check the validity of the response. This is because ξ with $G^{\xi} U^N \equiv E D^{\eta_j'} \pmod{N^2}$ is uniquely decrypted from $E D^{\eta_j'} \bmod N^2$ by $\mathcal{B}$, and we now consider only the case where commitment C is uniquely opened to E.

If ξ decrypted from $E D^{\eta_j'} \bmod N^2$, (η_j' is a challenge to $\mathtt{com}_{(\eta_1, \ldots, \eta_{j-1})}$) does not pass the verification, any path, $(\eta_1, \ldots, \eta_{j-1}, \eta_j', *, \ldots, *)$ with prefix $(\eta_1, \ldots, \eta_{j-1}, \eta_j')$ is 'lose' (i.e., $b_j \leftarrow 0$ is labelled as η_j' for any path $(\eta_1, \ldots, \eta_{j-1}, \eta_j', *, \ldots, *)$), since no valid response to challenge η_j' can be replied to at any step in path $(\eta_1, \ldots, \eta_{j-1}, \eta_j', *, \ldots, *)$, where $*$ can be any possible challenge value. If B passes the verification, we call η_j' 'survive'.

Let $\epsilon_{j-1,(\eta_1,\ldots,\eta_{j-1})}$ be the ratio of the j-th responses η_j to $\mathtt{com}_{(\eta_1,\ldots,\eta_{j-1})}$ that leads to 'survive' among all j-th responses $\mathtt{com}_{(\eta_1,\ldots,\eta_{j-1})}$). We call path $(\eta_1,\eta_2,\ldots,\eta_{q_S})$ *heavy* if $(\epsilon_{0,\perp}\cdots\epsilon_{q_S-1,(\eta_1,\ldots,\eta_{q_S-1})}) \geq \epsilon^+/2$. This leads to the following claim:

Claim. At least half of all 'win' paths are heavy.

This claim can be proven in a manner similar to that of the claim in the proof of Theorem 4.

We then obtain the success probability analysis similar to that in the proof of Theorem 4: If $\mathcal{B}$ tests $12\ln(q_S+1)/\epsilon^+$ challenges for each commitment, the probability that $\mathcal{B}$ can compute (m,t,s), is at least $(1-(q_S+1)^{-2})$. Overall, $\mathcal{B}$ breaks the q_S-unforgeability of the basic signature scheme with probability at least $(1-1/(q_S+1))\epsilon^+/2$ under the condition that $\mathcal{B}$ tests random challenges at most $12q_S\ln(q_S+1)/\epsilon^+$ times.

The above-mentioned evaluation is based on ϵ^+ under the fixed randomness of $\mathcal{G}$, $\mathcal{U}^*$ (here $\mathcal{A}$) and $\mathcal{S}$ except $\mathcal{S}$'s random challenges. Finally, over the whole random space, $\mathcal{B}$ breaks the q_S-unforgeability of the basic signature scheme with probability at least $(1-1/(q_S+1))\epsilon^*/8$ under the condition that $\mathcal{B}$ tests random challenges at most $24q_S\ln(q_S+1)/\epsilon^*$ times in total, where $\epsilon^* = \epsilon - \mu$.

Combining this result with Theorem 2, $\mathcal{B}$ breaks the (q_S,t',ϵ')-2SDH assumption

$$\epsilon' = \frac{1-1/(q_S+1)}{16}\cdot(\epsilon-\mu), \quad \text{and} \quad t' = t + O\Big(\frac{24q_S\ln(q_S+1)}{\epsilon-\mu}\cdot T\Big),$$

where T is the time for $\mathcal{B}$'s test, i.e., for decrypting Paillier encryption and checking the verification equations.

Since ϵ is non-negligible and μ is negligible in n, polynomial-time algorithm $\mathcal{B}$ breaks 2SDH assumption with non-negligible probability, which contradicts the assumption of this theorem. $\dashv$

9 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 [26], verifiably encrypted signatures, anonymous credentials and chameleon hash signatures. (The full paper version presents restrictive (partially) blind signatures based on our (partially) blind signatures.)

10 (Partially) Blind Signatures from the Waters Scheme

10.1 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 [4].
   (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 Hovav Shacham for his information on an attack on the original 2SDH assumption, Jun Furukawa for his suggestion on the security reduction to the SDH assumption in Theorem 1, and the anonymous reviewers of TCC 2006 for their invaluable comments and suggestions.

References

1. Abdalla, M., Namprempre, C., and Neven, G., On the (im)possibility of blind message authentication codes, CT-RSA 2006, LNCS 3860, pp. 262-279, Springer-Verlag (2006)
2. Abe, M., A Secure Three-Move Blind Signature Scheme for Polynomially Many Signatures, Eurocrypt'01, LNCS 2045, pp.136-151, Springer-Verlag (2001).
3. Abe, M. and Fujisaki, E., How to Date Blind Signatures, Asiacrypt'96, LNCS 1163, pp.244-251, Springer-Verlag (1996).
4. Abe, M. and Okamoto, T., Provably Secure Partially Blind Signatures, Crypto'00, LNCS 1880, pp.271-286, Springer-Verlag (2000).
5. 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).
6. 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).
7. Boneh, D. and Boyen, X., Short Signatures Without Random Oracles, Crypto'04, LNCS, Springer-Verlag (2004).
8. Boneh, D., Lynn, B. and Shacham, H., Short Signatures from the Weil Pairing, Asiacrypt'01, LNCS, Springer-Verlag (2001).
9. 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).
10. 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).
11. Camenisch, J. and Lysyanskaya, A., A signature scheme with efficient protocols, Security in communication networks, LNCS 2576, pp.268-289, Springer-Verlag (2002).
12. Camenisch, J. and Shoup, V., Practical verifiable encryption and decryption of discrete logarithms, Crypto'03, LNCS, pp. 126-144. Springer-Verlag (2003).
13. Camenisch, J. and Lysyanskaya, A., Signature Schemes and Anonymous Credentials from Bilinear Maps, Crypto'04, LNCS, Springer-Verlag (2004)
14. Chaum, D., Blind signatures for untraceable payments, Crypto'82, pp. 199-203. Plenum Press (1983).
15. 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).
16. Cramer, R. and Shoup, V., Signature schemes based on the strong RSA assumption, 6th ACM CCS, pp. 46-52. ACM press (1999).
17. Damgård, I., Efficient Concurrent Zero-Knowledge in the Auxiliary String Model, Eurocrypt'00, LNCS 1807, pp.418-430, Springer-Verlag (2000).
18. Diffie, W. and Hellma, M.E., New directions in cryptography, IEEE Trans. on Information Theory, IT-22(6), pp.644-654 (1976).
19. Fiat, A. and Shamir, A., How to prove yourself: Practical solution to identification and signature problems, Crypto'86, LNCS 263, Springer-Verlag (1987).
20. Fischlin, M., The Cramer-Shoup strong-RSA signature scheme revisited, PKC 2003, LNCS 2567, Springer-Verlag (2003).
21. Furukawa, J. and Imai, H.: An Efficient Group Signature Scheme from Bilinear Maps. ACISP 2005, LNCS 3574, pp.455-467 (2005).
22. 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).
23. 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).
24. Juels, A., Luby, M. and Ostrovsky, R., Security of blind digital signatures, Crypto'97, LNCS 1294, pp. 150-164, Springer-Verlag (1997).
25. Kiayias, A. and Zhou, H., Two-Round Concurrent Blind Signatures without Random Oracles, IACR Cryptology ePrint Archive, 2005/435 (2005)
26. Makita, T., Manabe, Y. and Okamoto, T., Short Group Signatures with Efficient Flexible Join, Manuscript (2005).

27. Mitsunari, S., Sakai, R. and Kasahara, M., A New Traitor Tracing, IEICE Trans. E-85-A, 2, pp. 481-484 (2002).
28. Naor, M. and Yung, M., Universal one-way hash functions and their cryptographic applications, 21st STOC, pp. 33-43, ACM (1989).
29. Okamoto, T., Efficient Blind and Partially Blind Signatures Without Random Oracles, TCC'06, LNCS, Springer-Verlag (2006).
30. Pointcheval, D., Strengthened security for blind signatures, Eurocrypt'98, LNCS, pp.391-405, Springer-Verlag (1998).
31. Pointcheval, D. and Stern, J., Provably secure blind signature schemes, Asiacrypt'96, LNCS, Springer-Verlag (1996).
32. Pointcheval, D. and Stern, J., New blind signatures equivalent to factorization, ACM CCS, pp. 92-99. ACM Press (1997).
33. Pointcheval, D. and Stern, J., Security arguments for digital signatures and blind signatures, Journal of Cryptology, 13, 3, pp.361-396, Springer-Verlag (2000).
34. Schnorr, C.P., Security of Blind Discrete Log Signatures against Interactive Attacks, ICICS'01, LNCS 2229, pp.1-12, Springer-Verlag (2001).
35. Shacham, H., Personal communication (2006)
36. Rompel, J., One-way functions are necessary and sufficient for secure signatures, STOC, pp.387-394, ACM (1990).
37. Waters, B., Efficient Identity-Based Encryption Without Random Oracles, Eurocrypt'05, LNCS 3494, pp. 114-127, Springer-Verlag (2005).
38. 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.