A New Certificateless Blind Signature Scheme

Sangeetha Jose*, Akash Gautam, and C Pandu Rangan
Indian Institute of Technology (IIT) Madras, Chennai-36, Tamilnadu, India
{sangeethajosem, akash.gautam24, prangan55}@gmail.com

Abstract

Blind signatures have key role in real world applications like e-cash, e-voting etc. The first blind signature was proposed by Chaum under public key infrastructure(PKI) model. The inherent problem in PKI is the certificate management which is overcome by identity(ID) based system. The ID based system is susceptible to key escrow problem. By removing the inherent problems of both PKI and ID based cryptosystems, Al Riyami et al. proposed a new cryptosystem called certificateless cryptosystem. Certificateless blind signature overcomes inherent key escrow problem in identity based blind signatures and does not require expensive certificates as in the public key infrastructure. Even though different certificateless blind signatures are proposed in the literature, rigorous formal proof is absent for all the proposals. Therefore in this paper we propose a new efficient provably secure certificateless blind signature scheme whose security can be proven to be equivalent to solving computational Diffie-Hellman (CDH) and chosen-target CDH problem in the random oracle model. As per our knowledge, our scheme is the only certificateless blind signature scheme which is proven to be strongly unforgeable and satisfies blindness property.

Keywords: certificateless blind signature scheme, unforgeability, blindness

1 Introduction

Certificateless cryptosystem overcomes the shortcomings of both public key infrastructure(*PKI*) and identity(*ID*) based cryptosystems. Al-Riyami et al. [1] proposed this new system in 2003 which does not require any certificates as in *PKI* based cryptosystem and also surmounts the key escrow problem of *ID* based cryptosystems. Certificateless cryptosystem also requires a trusted third party called key generation center(*KGC*). But *KGC* generates only some partial information (partial private key is computed by using identity of the user) for making the private key of the user. The private key of a user is generated by the partial private key from *KGC* and the secret value from the user. Therefore *KGC* alone cannot generate private key and thus avoids key escrow problem. User binds his secret value to his public key. Due to the lack of public key authentication, we need to assume that an adversary in the certificateless system can replace the user's public key with another value of his own choice, which is known as Type I adversary($\mathscr{A}_I$). Thus Type I represents a malicious third party. Type II adversary($\mathscr{A}_{II}$) represents a malicious *KGC* in which we assume that $\mathscr{A}_{II}$ can change *KGC*'s master secret key.

Related Work: Certificateless signature scheme was first proposed and constructed by Al-Riyami et al. [1]. Zhang et al. [2] proposed a certificateless signature scheme using pairings in elliptic curve groups. Liu et al. [3] came up with a certificateless signature scheme in the standard model. Huang et al. [4] elaborated new constructions for the security model of the certificateless signatures.

The idea of blind signature was put forward by David Chaum [5] in which the content of the message is blinded before it is signed. The blind signature allows the user to get a signature without giving any information about the message to the signer and the signer cannot tell which session of the signing protocol corresponds to which message [6]. The provable secure design for blind signature was proposed

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, volume: 5, number: 1, pp. 122-141
*Corresponding author: Tel: +92-(0)9447285935, She is currently working as an Assistant Professor in GEC Idukki

by Pointchevel et al. [7] in which they defined the security proof for blind signatures with an application to electronic cash. Juel et al. [8] projected first complexity based proof of security for blind signature. The properties of blind signatures are blindness and unforgeability. Therefore it has an eminent role in different cryptographic applications like e-voting, digital cash etc. Due to its extensive use, a lot of *PKI* based blind signature schemes [9, 10, 11, 12, 13, 14] as well as *ID* based blind signature schemes [15, 16] are evolved which is proven to be secure either in random oracle or in standard model. Generic construction of *ID* based blind signature is given by Galindo et al. [17].

Motivation: Due to the drawbacks of *PKI* and *ID* based schemes, certificateless blind signature schemes provide effective alternate for real world applications. Blind signatures are widely used in a number of cryptographic applications where signer has to authenticate a message for the user while maintaining the privacy of the user's message. Blind signatures have a crucial role in real world applications like e-vote, e-cash etc. Integrity of the e-voting requires that each ballot has to be certified by an election authority without learning voter's selection. Here we need to maintain the privacy of user's message(i.e, vote), at the same time it has to be authenticated(signed) by the authority. If the blind signature is certificateless, it overwhelms the drawbacks of *PKI* and *ID* based blind signatures since it does not require certificate management as well as does not have key escrow problem. Different certificateless blind signatures($\mathscr{CLBS}$) are available in the literature [18, 19, 20, 21]. These existing schemes claim the security without giving explicit mathematical proofs. Hence we construct a new certificateless blind signature scheme and prove it secure under computational Diffie-Hellman (CDH) and chosen-target CDH assumptions. Our approach is based on existing blind signature scheme by Boldyreva [9] and on the generic construction of identity based blind signature by Galindo et al. [17]. We use ideas of the aforementioned papers effectively and constructed a new efficient certificateless blind signature.

Organization of the Paper: Section 2 explains the preliminary concepts that discusses about bilinear pairing and also describes the hardness assumptions which help to prove the security of the proposed scheme. Section 3 gives the definitions of certificateless blind signature and its security notions. Section 4 discusses proposed certificateless blind signature scheme and its the proof of security elaborates in section 5. The paper concludes in section 6.

2 Preliminaries

2.1 Bilinear Pairing

Let $\mathbb{G}$ be a multiplicative cyclic group generated by g_1, with prime order q and $\mathbb{G}_1$ also be a multiplicative cyclic group of the same prime order q. A bilinear pairing is a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ with the following properties.

1. *Bilinearity*: For all $g_1, g_2, g_3 \in \mathbb{G}$
   - $e(g_1.g_2, g_3) = e(g_1, g_3).e(g_2, g_3)$
   - $e(g_1, g_2.g_3) = e(g_1, g_2).e(g_1, g_3)$
   - $e({g_1}^a, {g_2}^b) = e(g_1, g_2)^{ab}$ where $a, b \in_R \mathbb{Z}_q^*$
2. *Non-degeneracy*: There exist $g_1, g_2 \in \mathbb{G}$ such that $e(g_1, g_2) \neq I_{\mathbb{G}_1}$ where $I_{\mathbb{G}_1}$ is the identity element of $\mathbb{G}_1$.
3. *Computability*: There exists an efficient algorithm to compute $e(g_1, g_2)$ for all $g_1, g_2 \in \mathbb{G}$.

Security proof of the proposed scheme is based on computational Diffie-Hellman(CDH) and chosen-target computational Diffie-Hellman(Ct-CDH) assumptions.

2.2 Computational Diffie-Hellman(CDH) Problem and Assumption

$\mathbb{G}$ is a multiplicative cyclic prime order group with g as a generator, CDH problem states that given (g, g^a, g^b), we should obtain g^{ab}, where $g \in \mathbb{G}$ and $a, b \in_R \mathbb{Z}_q^*$.

Definition 1. *: (CDH Assumption): The advantage of any probabilistic polynomial time algorithm $\mathscr{A}$ in solving the CDH problem in $\mathbb{G}$ is defined as*
$Adv_{\mathscr{A}}^{CDH} = Prob[g^{ab} \leftarrow \mathscr{A}(g, g^a, g^b) \mid g \in \mathbb{G} \text{ and } a, b \in_R \mathbb{Z}_q^*]$
The Computational Diffie-Hellman(CDH) assumption is that, for any probabilistic polynomial time algorithm $\mathscr{A}$, the advantage $Adv_{\mathscr{A}}^{CDH}$ is negligibly small.

2.3 Chosen-target Computational Diffie-Hellman(Ct-CDH) Problem and Assumption

Boldyreva [9] proposed chosen-target computational Diffie-Hellman(Ct-CDH) problem and assumption as follows.

Definition 2. *: (Chosen-target CDH Problem and Assumption): Let $\mathbb{G}$ be a cyclic multiplicative group of a prime order q with generator g. Let x be a random element of ${\mathbb{Z}_q}^*$ and let $y = g^x$. Let H be a random instance of a hash function family $\{0,1\}^* \rightarrow \mathbb{G}^*$. The adversary $\mathscr{A}$ is given (q, g, H, y) and has access to the target oracle $\mathscr{T}_o$ that returns random points $z_i \in \mathbb{G}$ and a helper oracle $(.)^x$. Let q_t and q_h be the number of queries made to the target oracle and helper oracle respectively. The advantage of the adversary attacking the chosen-target CDH problem $Adv_{\mathscr{A}}^{Ct-CDH}$ is defined as the probability of adversary $\mathscr{A}$ to output a set V of, say, l pairs $\{(v_1, j_1), ...(v_l, j_l)\}$, where for all $1 \leq i \leq l, \exists 1 \leq j_i \leq q_t$ such that $v_i = z_{j_i}^x$, all v_i are distinct and $q_h < q_t$.*
The chosen-target CDH assumption states that there is no polynomial-time adversary $\mathscr{A}$ with non-negligible $Adv_{\mathscr{A}}^{Ct-CDH}$.

If the adversary makes one query to the target oracle then the chosen-target CDH assumption is equivalent to the standard CDH assumption. The chosen-target CDH assumption is hard for all groups where standard CDH problem is hard.

3 Certificateless Blind Signatures

3.1 Identity Based Blind Signatures

Definition 3. *: An identity based blind signature scheme ($\mathscr{IDBS}$) is defined as a 4-tuple of the following probabilistic polynomial-time algorithms $\Pi_{\mathscr{IDBS}}$ =($Setup_{\mathscr{IDBS}}$, $KeyExtract_{\mathscr{IDBS}}$, $Issue_{\mathscr{IDBS}}$, $Verify_{\mathscr{IDBS}}$), such that:*

- ***Setup*** $_{\mathscr{IDBS}}$*: This algorithm takes the security parameter 1^κ and generates system parameters called params, along with a master key pair (MSK, MPK) where MSK is the master secret key and MPK is the master public key of the trusted party called key generation center (KGC).*
- ***KeyExtract*** $_{\mathscr{IDBS}}$*: This is the key extraction algorithm run by KGC, which uses master secret key (MSK) and user's unique identity ID $\in \{0,1\}^*$ that may be his email id or some unique information, as input. It returns a secret key SK_{ID} corresponding to the one with identity ID.*
- ***Issue*** $_{\mathscr{IDBS}}$*: This is a joint interactive protocol between a user U and a signer A with identity ID_A. It includes three sub phases called Blind, Sign and UnBlind. In the Blind phase, user U will blind the message m by using some randomness and send the blind message m' to signer A. In the Sign phase, signer A will put sign on m' by using his private key SK_A and it ouputs blind signature σ'. Randomness will be removed in the UnBlind phase by the user U and obtains signature σ on m.*

- ***Verify*** $_{\mathcal{IDBS}}$*: This algorithm takes the master public key (MPK), system parameters (params), signer's public key (PK_A) and a message/signature pair (m,σ) as inputs and it outputs either valid or invalid. The output is valid if the signature is valid with respect to master public key (MPK), the signer's public key (PK_A) and system parameters (params); else outputs invalid.*

3.2 Certificateless Blind Signatures

Since identity based based blind signature is susceptible to *key escrow* problem, now a days certificateless schemes have more significance. This is because *KGC* contributes only partial information for the secret(private) key of the user.

Definition 4. *: A certificateless blind signature scheme($\mathcal{CLBS}$) is defined as a 7-tuple of the following probabilistic polynomial-time algorithms $\Pi_{\mathcal{CLBS}}$ =(Setup$_{\mathcal{CLBS}}$, Partial-Private-Key-Extract$_{\mathcal{CLBS}}$, Set-Secret-Value$_{\mathcal{CLBS}}$, Set-Public-Key$_{\mathcal{CLBS}}$, Set-Private-Key$_{\mathcal{CLBS}}$, Issue$_{\mathcal{CLBS}}$, Verify$_{\mathcal{CLBS}}$) such that:*

- ***Setup*** $_{\mathcal{CLBS}}$*: This algorithm is run by key generation center (KGC), which takes the security parameter 1^{κ} as input and generates system parameters called params, along with a master key pair (MSK, MPK) of the KGC who helps to generate secret key for the user.*

- ***Partial-Private-Key-Extract*** $_{\mathcal{CLBS}}$*: Partial key extraction algorithm is performed by KGC, which takes user's unique identity $ID \in \{0,1\}^*$ that may be his email id or some unique information, as input. KGC signs on the ID by using his master secret key, MSK and generates the output which is the partial private key of the user, D_{ID}.*

- ***Set-Secret-Value*** $_{\mathcal{CLBS}}$*: This algorithm takes params and user's identity ID as inputs and generates the secret value, α_{ID}. This algorithm is supposed to be run by each user in the system.*

- ***Set-Public-Key*** $_{\mathcal{CLBS}}$*: This algorithm takes system parameters (params), identity (ID) and secret value (α_{ID}) and outputs the public key, PK_{ID}.*

- ***Set-Private-Key*** $_{\mathcal{CLBS}}$*: This algorithm takes the public key (PK_{ID}), secret value (α_{ID}) and its partial private key (D_{ID}) as inputs. It outputs the private key SK_{ID} for the identity ID.*

- ***Issue*** $_{\mathcal{CLBS}}$*: This is a joint interactive protocol between a user U and a signer A with identity ID_A. It includes three sub phases called Blind, Sign and UnBlind. In the Blind phase, user U will blind the message m by using some randomness and send the blind message m' to signer A. In the Sign phase, signer A will put sign on m' by using his private key SK_A and it outputs blind signature σ'. Randomness will be removed in the UnBlind phase by the user U and outputs certificateless signature σ for m.*

- ***Verify*** $_{\mathcal{CLBS}}$*: This algorithm takes the the master public key (MPK), system parameters (params), signer's public key (PK_A) and a message/signature pair (m,σ) as inputs. It outputs valid or invalid. The output is valid if the signature is valid with respect to master public key MPK, the signer's public key PK_A and system parameters params; else the output is invalid.*

3.3 Security Notions for Certificateless Blind Signatures

Security notions for certificateless blind signatures deal with the blind signature properties unforgeability and blindness. It is an interactive simulation model between challenger and adversary.

3.3.1 Unforgeability

The security of the certificateless scheme can be analysed by considering two types of adversaries[1]. Due to the uncertified nature of the public key, an adversary in the certificateless system can replace user's public key with another value of his own choice, which is known as Type I adversary. The second type of adversary (Type II) represents a malicious *KGC* who generates partial private key for the users. Here, the adversary is equipped with the *KGC*'s master secret key, but cannot replace any user's public key. Our definition of unforgeability for certificateless blind signatures is adapted from the concept of $(l, l^{'})$ unforgeability introduced in [8] for *PKI* based blind signatures.
The proof technique that we followed is the proof technique used in [2], [4]. Let $\mathscr{A}_I$ and $\mathscr{A}_{II}$ are Type I attacker and Type II attacker respectively. There are two games *Game I* and *Game II* in which $\mathscr{A}_I$ and $\mathscr{A}_{II}$ interact with the challenger $\mathscr{C}$ in their respective games. Each adversary is polynomially bounded with respect to the security parameter κ. This means that the computational power, the number of interactions with the challenger and the time available for generating its output are at most polynomial in terms of κ. We say that a $\mathscr{CLBS}$ scheme is unforgeable if the success probability of both $\mathscr{A}_I$ and $\mathscr{A}_{II}$ is negligible. Let Π be the $\mathscr{CLBS}$ scheme, then the description of the two games against the Type I adversary $\mathscr{A}_I$ and the Type II adversary $\mathscr{A}_{II}$ is as follows.
Game I*:* This game is performed between the challenger $\mathscr{C}$ and the Type I adversary $\mathscr{A}_I$ for a $\mathscr{CLBS}$ scheme Π as follows.

1. **Setup Phase:** Challenger $\mathscr{C}$ runs the setup algorithm $Setup_{\mathscr{CLBS}}$ with security parameter κ, which generates parameters (*params*). $\mathscr{C}$ gives *params* and *MPK* to the Type I adversary $\mathscr{A}_I$.

2. **Training Phase:** $\mathscr{A}_I$ can perform a polynomial number of queries to each of the following oracles provided by $\mathscr{C}$. The current query may depends on responses to the previous queries and hence it may be adaptive.

    - *ExtractPartSK Oracle(ID):* $\mathscr{A}_I$ requests $\mathscr{C}$ for the partial private key D_{ID} for a user with identity *ID*. $\mathscr{C}$ obtains D_{ID} by running the oracle *ExtractPartSK* by passing *ID* to the oracle as parameter and returns D_{ID} to $\mathscr{A}_I$. It keeps this information in the list L_{PartSK} which contains D_{ID} corresponds to identity *ID*.
    - *RequestPK Oracle(α_{ID},ID):* $\mathscr{A}_I$ requests the public key for a user with identity *ID*, $\mathscr{C}$ returns the public key PK_{ID} by running *RequestPK* oracle by providing α_{ID} which is a random value and *ID* as parameters. It keeps the information in the list L_{PK} which contains PK_{ID} and its corresponding α_{ID} for further use.
    - *ExtractFullSK Oracle(D_{ID},α_{ID},ID):* When $\mathscr{A}_I$ requests the full private key SK_{ID} for a user with identity *ID*, $\mathscr{C}$ first runs the oracle *ExtractPartSK* and obtains the partial private key D_{ID}. $\mathscr{C}$ also obtains secret value α_{ID} from L_{PK}. Now challenger can obtain SK_{ID} by running the *ExtractFullSK Oracle* by providing D_{ID} and α_{ID} as parameters and sends it back to $\mathscr{A}_I$.
    - *ReplacePK Oracle(ID,$PK^{'}_{ID}$)*: $\mathscr{A}_I$ can replace the public key PK_{ID} of an entity with some value $PK^{'}_{ID}$. Since $\mathscr{A}_I$ can replace the public key as $PK^{'}_{ID}$, $\mathscr{C}$ runs *ReplacePK Oracle* by providing ID and $PK^{'}_{ID}$ as parameters. The replaced value will be updated in the list L_{PK}.
    - *Issue Oracle($m^{'}$,ID,sv)*: $\mathscr{A}_I$ requests a blind signature $\sigma^{'}$ on a message $m^{'}$ with secret value $sv \in \{nil\} \cup \mathscr{S}$, where $\mathscr{S}$ is a set of valid secret value, for a party with identity *ID*. $\mathscr{C}$ runs *Issue Oracle* by passing *ID*, $m^{'}$ and *sv* as parameters. Here the secret value *sv* can be the original secret value α_{ID} chosen by the user with identity *ID*, or the secret value supplied by the adversary $\mathscr{A}_I$. If $sv = nil$, indicates that the secret value is the original secret value

α_{ID} chosen by the user ID and hence this oracle uses ID's the original secret value α_{ID} and partial private key D_{ID} to generate the signature $\sigma^{'}$ as output. If $sv \in \mathscr{S}$, indicates that sv is the secret value corresponding to the replaced public key by $\mathscr{A}_I$ and hence this oracle uses this sv and ID's partial private key D_{ID} to generate the signature $\sigma^{'}$. *Issue Oracle* outputs the blind signature $\sigma^{'}$ and $\mathscr{C}$ returns this $\sigma^{'}$ to $\mathscr{A}_I$. Here $\mathscr{A}_I$ is a strong Type I adversary.

3. **Forgery Phase:** Let l be the number of such *Issue* queries that finished successfully. Eventually, the $\mathscr{A}_I$ outputs a list of $l^{'}$ tuples (m_i, ID_i, σ_i) where $1 \leq i \leq l^{'}$. We say that $\mathscr{A}_I$ wins the game if,

   - $l < l^{'}$
   - $Verify_{\mathscr{CLBS}}(MPK, m_i, ID_i, PK_{ID_i}, \sigma_i) \rightarrow 1$, which means all output signatures must be valid with respect to the master public key MPK and the corresponding public key PK_{ID_i}, where the latter may have been replaced by $\mathscr{A}_I$.
   - The triples (m_i, ID_i, σ_i) included in the output list are all distinct.
   - $\mathscr{A}_I$ did not ask *ExtractPartSK Oracle(ID_i)* and *ExtractFullSK Oracle(ID_i)* queries for any of the identities ID_i in the output list.

We can say that the adversary $\mathscr{A}_I$ which satisfies the above conditions is an $(l, l^{'})$ adversary. We define $Adv^{\Pi}_{\mathscr{A}_I}$ to be the success probability that $\mathscr{A}_I$ wins in the above game.

Game II: This game is performed between the challenger $\mathscr{C}$ and the Type II adversary $\mathscr{A}_{II}$ for a $\mathscr{CLBS}$ scheme Π. The Type II adversary represents a malicious *KGC* who generates partial private key for the users. Here, the adversary is equipped with the *KGC*'s master secret key. The game is as follows.

1. **Setup Phase:** Challenger $\mathscr{C}$ runs the setup algorithm $Setup_{\mathscr{CLBS}}$ with security parameter κ, which generates system parameters (*params*) and a master key pair(MPK, MSK). $\mathscr{C}$ gives *params* as well as the master key pair to the Type II adversary $\mathscr{A}_{II}$. Note that $\mathscr{A}_{II}$ gets master secret key (MSK) since it represents a malicious *KGC*.

2. **Training Phase:** $\mathscr{A}_{II}$ can perform a polynomial number of queries to each of the following oracles provided by $\mathscr{C}$. The current query may depends on responses to the previous queries and hence it may be adaptive. Note that the partial private key D_{ID} can be computed by both $\mathscr{C}$ and $\mathscr{A}_{II}$ because $\mathscr{A}_{II}$ also holds the master secret key MSK.

   - *ExtractFullSK Oracle(D_{ID},α_{ID},ID):* When $\mathscr{A}_{II}$ requests the full private key SK_{ID} for a user with identity ID, $\mathscr{C}$ first runs the oracle *ExtractPartSK* and obtains the partial private key D_{ID}. $\mathscr{C}$ also obtains secret value α_{ID} from L_{PK}. Now challenger can obtain SK_{ID} by running the *ExtractFullSK Oracle* by providing D_{ID} and α_{ID} as parameters and sends it back to $\mathscr{A}_{II}$.
   - *RequestPK Oracle(α_{ID},ID):* $\mathscr{A}_{II}$ requests the public key for a user with identity ID, $\mathscr{C}$ returns the public key PK_{ID} by running *RequestPK* oracle by providing α_{ID} which is a random value and ID as parameters. It keeps this information in the list L_{PK} which contains PK_{ID} and its corresponding α_{ID} for further use.
   - *Issue Oracle($m^{'}$,ID):* $\mathscr{A}_{II}$ requests a blind signature $\sigma^{'}$ on a message $m^{'}$ for a party with identity ID. Challenger $\mathscr{C}$ provides the full secret key SK_{ID}(by running *ExtractFullSK* Oracle), $m^{'}$ and ID as parameters and computes signature $\sigma^{'}$. Note that the corresponding public key PK_{ID} is not replaced by $\mathscr{A}_{II}$, so $\mathscr{C}$ can compute all SK_{ID}.

3. **Forgery Phase:** Let l be the number of such *Issue* queries that finished successfully. Eventually, the $\mathscr{A}_{II}$ outputs a list of $l^{'}$ tuples (m_i, ID_i, σ_i) where $1 \leq i \leq l^{'}$. We say that $\mathscr{A}_{II}$ wins the game if,

- $l < l^{'}$
- $Verify_{\mathscr{CLBS}}(MPK, m_i, ID_i, PK_{ID_i}, \sigma_i) \rightarrow 1$, which means all output signatures must be valid with respect to the master public key MPK and the corresponding public key PK_{ID_A}.
- The triples (m_i, ID_i, σ_i) included in the output list are all distinct.
- $\mathscr{A}_{II}$ did not ask *ExtractFullSK Oracle(ID_i)* queries for any of the identities ID_i in the output list.

We can say that the adversary $\mathscr{A}_{II}$ which satisfies the above conditions is an $(l, l^{'})$ adversary. We define $Adv^{\Pi}_{\mathscr{A}_{II}}$ to be the success probability that $\mathscr{A}_{II}$ wins in the above game.

Definition 5. *A certificateless blind signature scheme $\Pi_{\mathscr{CLBS}}$ is strongly unforgeable against $(l, l^{'})$ adversary under chosen message attacks, if for any polynomially bounded, probabilistic adversaries $\mathscr{A}_I$ and $\mathscr{A}_{II}$, the advantage of winning the games(Game I and Game II) is negligible in terms of the security parameter κ. In other words, $Adv^{\Pi_{\mathscr{CLBS}}}_{\mathscr{A}_I} < \varepsilon$ and $Adv^{\Pi_{\mathscr{CLBS}}}_{\mathscr{A}_{II}} < \varepsilon$ where ε is a negligible function in κ.*

3.3.2 Blindness

Blindness property indicates that the signer should not get any information about the message which he signs for a particular user during the signing process. The message m is blinded and submitted by the user for getting the signature. Unlinkability property which prevents the signer from linking the blinded message supplied during the protocol to an unblinded version which is later called upon to verify. Security notion of blindness is described with respect to a game played between a challenger $\mathscr{C}$ and an adversary $\mathscr{A}_{Blind}$. Here $\mathscr{A}_{Blind}$ represents a malicious signer who tries to distinguish between messages m_0 and m_1(chosen by $\mathscr{A}_{Blind}$ himself) which will be blindly signed by an interactive signing algorithm with the user. Game for blindness is as follows.

1. **Setup Phase:** $\mathscr{C}$ runs the setup algorithm $Setup_{\mathscr{CLBS}}$ with security parameter κ, and generates parameters (*params*). $\mathscr{C}$ provides *params* and MPK to the $\mathscr{A}_{Blind}$.

2. **Training Phase:** $\mathscr{A}_{Blind}$ may perform polynomial number of queries to the following oracle provided by $\mathscr{C}$.

   - *ExtractPartSK Oracle(ID):* $\mathscr{A}_{Blind}$ requests $\mathscr{C}$ for the partial private key D_{ID} for a user with identity ID. $\mathscr{C}$ obtains D_{ID} by running the oracle *ExtractPartSK* by passing ID to the oracle as parameter and returns D_{ID} to $\mathscr{A}_{Blind}$. It keeps this information in the list L_{PartSK} which contains D_{ID} corresponds to identity ID.

   $\mathscr{A}_{Blind}$ does not need to query any other oracles because secret value(α_{ID}), public key(PK_{ID}) and secret key(SK_{ID}) can be generated by himself. That is, a malicious signer can be modelled who can choose all the system values by himself, except for those which depend up on master secret key.

3. **Challenge Phase:** $\mathscr{A}_{Blind}$ chooses an identity ID^*, two messages m_0, m_1 and give to the challenger. $\mathscr{C}$ randomly chooses a bit $b \in_R \{0,1\}$. The interactive signing protocol is executed for two instances between user($\mathscr{C}$) and signer($\mathscr{A}_{Blind}$) in a concurrent manner. Let Issue$_{\mathscr{CLBS}}$(params,m_b, ID^*, SK_{ID^*}) and Issue$_{\mathscr{CLBS}}$(params,m_{1-b}, ID^*, SK_{ID^*}) be the two instances. When the execution of both instances completed $\mathscr{A}_{Blind}$ will get the tuple $(\sigma^*_b, \sigma^*_{1-b})$. If any of them aborts(user or signer), $\mathscr{A}_{Blind}$ gets the tuple $(\perp, \perp)$.

Finally $\mathscr{A}_{Blind}$ produces a guess bit $b^{'}$. If $b = b^{'}$, the adversary $\mathscr{A}_{Blind}$ is said to win the game. We define $Adv^{\Pi}_{\mathscr{A}_{Blind}}$ to be the success probability that $\mathscr{A}_{Blind}$ wins in the above game and is equal to $|Pr[b = b^{'}] - \frac{1}{2}|$.

Definition 6. *A certificateless blind signature scheme $\Pi_{\mathscr{CLBS}}$ has blindness property, if for any polynomially bounded probabilistic adversary $\mathscr{A}_{Blind}$, the advantage of winning the game is negligible in terms of the security parameter κ. In other words, $Adv^{\Pi_{\mathscr{CLBS}}}_{\mathscr{A}_{Blind}} < \varepsilon$ where ε is a negligible function in κ.*

4 The Proposed Scheme

In this section, we propose an efficient certificateless blind signature (*CLBS*) scheme and illustrate a concrete security proof in the following section. Proposed certificateless blind signature scheme comprises the following algorithms.

- **Setup**$_{\mathscr{CLBS}}(1^{\kappa})$**:** On input the security parameter κ, *Setup* phase is as follows.

  (a) Select a pairing $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$ where $\mathbb{G}$ and $\mathbb{G}_1$ are multiplicative prime order group in q.

  (b) Choose a random integer $x \in_R \mathbb{Z}_q^*$ and a generator $g \in \mathbb{G}$. Compute $P_{pub} = g^x$.

  (c) Select cryptographic hash functions, $H_1 : \{0,1\}^* \rightarrow \mathbb{G}$ and $H_2 : \{0,1\}^* \times \mathbb{G} \rightarrow \mathbb{G}$.

  Return master secret key, $MSK = x$, master public key, $MPK = P_{pub}$ and system parameters $params = \{e, \mathbb{G}, \mathbb{G}_1, q, g, P_{pub}, H_1, H_2\}$.

- **Partial-Private-Key-Extract**$_{\mathscr{CLBS}}$: On input params, master secret key $MSK = x$ and identity of user ID_A, key generation center(KGC) will generate the certificate as follows. Randomly choose $sk_A \in_R \mathbb{Z}_q^*$, compute g^{sk_A} and $cert_A = H_1(ID_A || g^{sk_A})^x$. Return partial private key $D_A = (cert_A, g^{sk_A}, sk_A)$.

- **Set-Secret-Value**$_{\mathscr{CLBS}}$: Choose $\alpha_A \in_R \mathbb{Z}_q^*$ and return α_A as user's secret value.

- **Set-Public-Key**$_{\mathscr{CLBS}}$: Compute $PK_A = g^{\alpha_A}$ and return the public key of the user PK_A.

- **Set-Private-Key**$_{\mathscr{CLBS}}$: Return the secret key of the user, $SK_A = (\alpha_A, D_A)$.

- **Issue**$_{\mathscr{CLBS}}$: Issue phase includes blinding, signing and unblinding algorithms.

  (a) Blinding: User U randomly choose $r \in_R \mathbb{Z}_q^*$ and computes $m' = H_2(m, PK_A)\ g^r$. User sends m' to the signer A.

  (b) Signing: Signer A computes
  - $\sigma_1^{'} = (m')^{\alpha_A}$
  - $\sigma_2^{'} = (m')^{sk_A}$
  - $\sigma_3^{'} = cert_A$
  - $\sigma_4^{'} = g^{sk_A}$

  Signer A sends back the tuple $\sigma^{'} = (\sigma_1^{'}, \sigma_2^{'}, \sigma_3^{'}, \sigma_4^{'})$ to user U.

  (c) Unblinding: User U first verifies whether $e(\sigma_3^{'}, g) = e(H_1(ID_A || \sigma_4^{'}), P_{pub})$.
  If the output is 'null' the algorithm finishes with $Out_U =$ '$fail$' and $Out_A =$ '$not\ completed$'.
  Else the user U calculates
  - $\sigma_1 = \sigma_1^{'}(PK_A)^{-r}$
  - $\sigma_2 = \sigma_2^{'}(\sigma_4^{'})^{-r}$

- $\sigma_3 = \sigma_3^{'}$
- $\sigma_4 = \sigma_4^{'}$ and returns signature, $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$.

- **Verify**$_{\mathscr{CLBS}}$: On input message m, identity ID_A, public Key PK_A and blind signature $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$, check whether

$$e(\sigma_1, g) \stackrel{?}{=} e(H_2(m, PK_A), PK_A) \tag{1}$$

$$e(\sigma_2, g) \stackrel{?}{=} e(H_2(m, PK_A), \sigma_4) \tag{2}$$

$$e(\sigma_3, g) \stackrel{?}{=} e(H_1(ID_A||\sigma_4), P_{pub}) \tag{3}$$

Output 'valid' if the all checks are correct; else output 'invalid'.
Correctness of first verification algorithm (1)

$$\begin{aligned}
e(\sigma_1, g) &= e(\sigma_1^{'}(PK_A)^{-r}, g) \\
&= e((m')^{\alpha_A}.(PK_A)^{-r}, g) \\
&= e((H_2(m, PK_A).g^r)^{\alpha_A}.(PK_A)^{-r}, g) \\
&= e((H_2(m, PK_A))^{\alpha_A}.(g^r)^{\alpha_A}.(PK_A)^{-r}, g) \\
&= e((H_2(m, PK_A))^{\alpha_A}.(g^{\alpha_A})^r.(g^{\alpha_A})^{-r}, g) \\
&= e((H_2(m, PK_A))^{\alpha_A}, g) \\
&= e(H_2(m, PK_A), g^{\alpha_A}) \\
&= e(H_2(m, PK_A), PK_A)
\end{aligned}$$

Correctness of second verification algorithm (2)

$$\begin{aligned}
e(\sigma_2, g) &= e(\sigma_2^{'}(\sigma_4^{'})^{-r}, g) \\
&= e((m')^{sk_A}.(\sigma_4^{'})^{-r}, g) \\
&= e((H_2(m, PK_A).g^r)^{sk_A}.(\sigma_4^{'})^{-r}, g) \\
&= e((H_2(m, PK_A))^{sk_A}.(g^r)^{sk_A}.(\sigma_4^{'})^{-r}, g) \\
&= e((H_2(m, PK_A))^{sk_A}.(g^{sk_A})^r.(g^{sk_A})^{-r}, g) \\
&= e((H_2(m, PK_A))^{sk_A}, g) \\
&= e(H_2(m, PK_A), g^{sk_A}) \\
&= e(H_2(m, PK_A), \sigma_4)
\end{aligned}$$

Correctness of the final verification algorithm (3)

$$\begin{aligned}
e(\sigma_3, g) &= e(cert_A, g) \\
&= e((H_1(ID_A||g^{sk_A}))^x, g) \\
&= e(H_1(ID_A||g^{sk_A}), g^x) \\
&= e(H_1(ID_A||\sigma_4), P_{pub})
\end{aligned}$$

5 Proof of Security

Proposed scheme is unforgeable with respect to the chosen message attack. We can also show that the scheme satisfies the computational blindness property.

5.1 Unforgeability

Theorem 1. *Proposed certificateless blind signature scheme $\Pi_{\mathscr{CLBS}}$ is $(l,l^{'})$ strongly unforgeable against chosen message attacks in the random oracle model, under the CDH and the chosen-target CDH assumptions.*

Proof: From the following Lemma 1 and Lemma 2 we can prove that the Theorem 1 is valid. Since neither CDH nor chosen-target CDH problem can be solved in polynomial time with non-negligible probability, it negates the existence of a Type I adversary ($\mathscr{A}_I$) or a Type II adversary ($\mathscr{A}_{II}$) who has at least non-negligible probability of success in the following games.

5.1.1 Security Analysis Against Type I Adversary

Lemma 1. *If there exists a Type I adversary $\mathscr{A}_I$ that succeeds in winning $(l,l^{'})$ strong unforgeability under chosen message attack with non-negligible probability in polynomial time in random oracle model, then either the computational Diffie-Hellman(CDH) problem or the chosen-target CDH problem can be solved with non-negligible probability in polynomial time.*

Proof:- Let $\mathscr{A}_I$ be a Type I adversary who has an advantage ε_{adv} in breaking the unforgeability property of the proposed scheme. We want to build a simulated algorithm using $\mathscr{A}_I$ which solves either the computational Diffie-Hellman(CDH) problem or the chosen-target CDH problem.
Let the CDH instance be (g,g^a,g^b) where $a,b \in \mathbb{Z}_q^*$ and $g \in \mathbb{G}$. The goal of challenger is to compute g^{ab}. Let the given chosen-target CDH instance be (g,g^{γ}) where $\gamma \in_R \mathbb{Z}_q^*$. A target oracle T_O outputting random elements z_i in $\mathbb{G}$ and helper oracle $H_O = (.)^{\gamma}$ is provided. Without loss of generality, we assume that the number of queries to H_1 oracle exceeds the number of queries to other oracles, since oracles simulating other stages make use of H_1 queries. Let the maximum number of H_1 queries be q_{H_1}. Also we assume that each oracle maintains a list of answered queries for consistent replies to the same query. Let these list be called $L_{H_1}, L_{H_2}, L_{PartSK}$ and L_{PK}.
We set $P_{pub} = g^a$ and params=$(e,\mathbb{G},\mathbb{G}_1,q,g,P_{pub},H_1,H_2)$. Simulation game will be as follows.

- H_1 **Oracle**($ID_i||\beta$): $\mathscr{A}_I$ queries challenger for H_1 value by inputting ID_i and β. Challenger will output the value as follows. Randomly choose j such that $1 \leq j \leq q_{H_1}$.
  If $i = j$,(at this point, we let $ID_i = ID^*$)
  if $\beta = g^{\gamma}$, return g^{r^*}, $r^* \in_R \mathbb{Z}_q^*$.
  else, return $g^{r^* b}$, $r^* \in_R \mathbb{Z}_q^*$.
  If $i \neq j$, challenger will first search for the entry corresponding to ID_i in L_{PartSK} and get g^{sk_i}. If the entry exists,
  if $\beta = g^{sk_i}$, return g^{r_i}, $r_i \in_R \mathbb{Z}_q^*$.
  else, return $g^{r_i b}$, $r_i \in_R \mathbb{Z}_q^*$.
  If the entry does not exist, choose $sk_i \in_R \mathbb{Z}_q^*$ and compute g^{sk_i}.
  if $\beta = g^{sk_i}$, return g^{r_i}, $r_i \in_R \mathbb{Z}_q^*$ and append the entry $< ID_i, g^{r_i a}, g^{sk_i}, sk_i >$ to L_{PartSK}.
  else, return $g^{r_i b}$, $r_i \in_R \mathbb{Z}_q^*$ and append the entry $< ID_i, g^{r_i a}, g^{sk_i}, sk_i >$ to L_{PartSK}.
  Append entry $< ID_i, \beta, H_1(ID_i||\beta) >$ to list L_{H_1}.

- H_2 **Oracle**(m_i, PK_i): On inputting m_i and PK_i, if there exists an entry $< m_i, PK_i, z_i >$ in L_{H_2} list, return z_i. Else, get a random element $z_i \in \mathbb{G}$ from the target oracle T_O, return z_i and append entry $< m_i, PK_i, z_i >$ to list L_{H_2}.

- **ExtractPartSK Oracle**(ID_i): If $ID_i = ID^*$, abort. Else if the corresponding entry $< ID_i, cert_i, g^{sk_i}, sk_i >$ exists in list L_{PartSK}, return $D_i = (cert_i, g^{sk_i}, sk_i)$. If the entry does not exist, choose $sk_i \in_R \mathbb{Z}_q^*$ and compute g^{sk_i}. Compute $cert_i = [H_1(ID_i || g^{sk_i})]^a = (g^{r_i})^a = (g^a)^{r_i} = P_{pub}^{r_i}$. Append $< ID_i, D_i >$ to list L_{PartSk} and return $D_i = (cert_i, g^{sk_i}, sk_i)$.

- **RequestPK Oracle**(ID_i): If there is no entry in the list L_{PK}, choose $\alpha_i \in_R \mathbb{Z}_q^*$ and output g^{α_i} as public key of ID_i. Append entry $< ID_i, \alpha_i, PK_i, 1 >$ to list L_{PK}. The entry 1 denotes that public key for ID_i has not been replaced.

- **ExtractFullSK Oracle**(ID_i): If $ID_i = ID^*$, abort. Else get the corresponding entries by calling $ExtractPartSK(ID_i)$ and $RequestPK(ID_i)$. Output full secret key as $SK_i = (\alpha_i, D_i)$.

- **ReplacePK Oracle**$(ID_i, PK_i^{'})$: If corresponding ID_i entry exists in L_{PK}, replace $< ID_i, \alpha_i, PK_i, 1 >$ with $< ID_i, -, PK_i^{'}, 0 >$. Else append an entry $< ID_i, -, PK_i^{'}, 0 >$. The entry 0 denotes that public key has been replaced.

- **Issue Oracle**(m', ID_i)**:**

  - If $ID_i \neq ID^*$, get corresponding entry $< ID_i, cert_i, g^{sk_i}, sk_i >$ from L_{PartSK}. Calculate:
    - $\sigma_2^{'} = (m^{'})^{sk_i}$
    - $\sigma_3^{'} = cert_i$
    - $\sigma_4^{'} = g^{sk_i}$

    Get corresponding entry $< ID_i, \alpha_i, PK_i, c >$ from L_{PK}. If $c = 0$, request corresponding secret key $\alpha_i^{'}$ from adversary for the replaced public key.
    Calculate $\sigma_1^{'} = (m^{'})^{\alpha_i}$ (if c=1), or $\sigma_1^{'} = (m^{'})^{\alpha_i^{'}}$ (if c=0). Then challenger outputs the tuple $\sigma^{'} = (\sigma_1^{'}, \sigma_2^{'}, \sigma_3^{'}, \sigma_4^{'})$ and gives to $\mathscr{A}_I$.
  - If $ID_i = ID^*$, get corresponding entry $< ID^*, g^{\gamma}, g^{r^*} >$ from L_{H_1}. Then calculate:
    - $\sigma_2^{'} = (m^{'})^{\gamma}$ [By querying helper oracle H_O]
    - $\sigma_3^{'} = cert^* = (g^{r^*})^a = (P_{pub})^{r^*}$
    - $\sigma_4^{'} = g^{\gamma}$

    Get the corresponding entry $< ID^*, \alpha^*, PK^*, c >$ from L_{PK}. If $c = 0$, request corresponding secret key $\alpha^{'^*}$ from adversary for the replaced public key.
    Calculate $\sigma_1^{'} = (m^{'})^{\alpha^*}$ (if c=1), or $\sigma_1^{'} = (m^{'})^{\alpha^{'^*}}$ (if c=0).
    Thus challenger outputs the tuple $\sigma^{'} = (\sigma_1^{'}, \sigma_2^{'}, \sigma_3^{'}, \sigma_4^{'})$ and gives to $\mathscr{A}_I$.

We can see that this simulation game is indistinguishable from the actual certificateless blind signature scheme for the view of adversary $\mathscr{A}_I$. Eventually, $\mathscr{A}_I$ engages in l successful runs of Issue oracle with the challenger and outputs $l^{'}$ tuples of (m_i, ID_i, σ_i) which are distinct and $l < l^{'}$. Since it is a Type I adversary, ExtractPartSK and ExtractFullSK oracles have not been queried for all $1 \leq i \leq l^{'}$. We can now analyse the following cases.

Case 1: In the adversary's output, there exists a pair (m_i, ID_i) such that no Issue query has been asked for ID_i. Let this pair be $(\bar{m}, \bar{ID})$ and the corresponding signature output be $\bar{\sigma} = (\bar{\sigma}_1, \bar{\sigma}_2, \bar{\sigma}_3, \bar{\sigma}_4)$. Since for $\bar{ID}$, neither ExtractPartSK nor ExtractFullSK oracles are queried and hence the adversary has no information about corresponding $g^{\bar{sk}}$. It is also chosen completely random during simulation and is independent

of any other parameters or protocols. Therefore, adversary can know about $g^{\bar{sk}}$ only with negligible probability. That is,

$$\Pr[\bar{\sigma}_4 = g^{\bar{sk}}]=\frac{1}{q}$$
$$\implies \Pr[\bar{\sigma}_4 \neq g^{\bar{sk}}]=(1-\frac{1}{q})$$
$$\implies \Pr[H_1(\bar{ID}||\bar{\sigma}_4) = g^{\bar{r}b}]=(1-\frac{1}{q}) \quad \text{[By definition of } H_1 \text{ oracle]}$$

Being a valid signature it satisfies verification algorithm (3) also.

$$e(\bar{\sigma}_3, g) = e(H_1(\bar{ID}||\bar{\sigma}_4), P_{pub})$$
$$\implies e(\bar{\sigma}_3, g) = e(g^{\bar{r}b}, g^a)$$
$$\implies \bar{\sigma}_3 = g^{\bar{r}ab}$$

$\bar{r}$ is chosen by the challenger during H_1 oracle query phase and so challenger can compute $(\bar{\sigma}_3)^{\frac{1}{\bar{r}}} = (g^{\bar{r}ab})^{\frac{1}{\bar{r}}} = g^{ab}$. Therefore, case 1 is equivalent to solve CDH problem.

Let Pr[Case 1 occurs] = τ. Assuming Type I adversary $\mathscr{A}_I$, who has an advantage ε_{adv} in breaking the unforgeability property, then,

$$Pr[CDH \text{ is solved}] \geq \varepsilon_{adv}.\tau.(1-\frac{1}{q}) \tag{4}$$

Case 2: In the adversary's output, there does not exist a pair (m_i, ID_i) such that no Issue query has been asked for ID_i. Therefore, all IDs in the adversary's output have appeared in some Issue query and hence the maximum number of output IDs is bounded by q_{H_1}.

Here we have $l < l^{'}$ and hence for at least one output we have $l_{ID_i} < l^{'}_{ID_i}$ where $l^{'}_{ID_i}$ is the number of valid output signatures with respect to ID_i and l_{ID_i} is the number of times adversary engaged in Issue protocol with respect to ID_i. Therefore, the probability that ID^* has this property is at least $\frac{1}{q_{H_1}}$ because adversary gains no information about the chosen ID^* and it is equally likely for each ID_i to have this property.

In case ID^* satisfies the above property, it means l_{ID^*} Issue protocol were successfully completed with respect to ID^* and hence helper oracle H_O was queried l_{ID^*} times. Let the $l^{'}_{ID^*}$ output tuples containing ID^* be $\{(m_t, ID^*, \sigma_t)\}$ where $1 \leq t \leq l^{'}_{ID^*}$ and $\sigma_t = (\sigma_{t1}, \sigma_{t2}, \sigma_{t3}, \sigma_{t4})$.

Case 2a: There exists a t such that $\sigma_{t4} \neq g^{\gamma}$. Being a valid signature, σ_t satisfies verification algorithm (3) and therefore,

$$e(\sigma_{t3}, g) = e(H_1(ID^*||\sigma_{t4}), P_{pub})$$
$$\implies e(\sigma_{t3}, g) = e(g^{r^*b}, g^a)$$
$$\implies \sigma_{t3} = g^{r^*ab}$$

r^* is chosen by the challenger during H_1 oracle query phase and so challenger can compute $(\sigma_{t3})^{\frac{1}{r^*}} = (g^{r^*ab})^{\frac{1}{r^*}} = g^{ab}$. Therefore, case 2a is equivalent to solve CDH problem.

Pr[Case 2 occurs] = $1-\tau$.

$\Pr[l_{ID^*} < l^{'}_{ID^*}] \geq \frac{1}{q_{H_1}}$

Let Pr[Case 2a occurs] = θ. Then,

$$Pr[CDH \text{ is solved}] \geq \varepsilon_{adv}.(1-\tau).(\frac{1}{q_{H_1}}).\theta \tag{5}$$

Case 2b: There does not exists a t such that $\sigma_{t4} \neq g^{\gamma}$. Hence for all $1 \leq t \leq l^{'}_{ID^*}$, $\sigma_{t4} = g^{\gamma}$. Being a valid signature, σ_t satisfies verification algorithm (2) and therefore,

$$e(\sigma_{t2}, g) = e(H_2(m_t, PK^*), \sigma_{t4})$$
$$\implies e(\sigma_{t2}, g) = e(H_2(m_t, PK^*), g^{\gamma})$$
$$\implies e(\sigma_{t2}, g) = e(z_t, g^{\gamma})$$
$$\implies \sigma_{t2} = z_t^{\gamma}$$

This holds for all $1 \leq t \leq l^{'}_{ID^*}$. So adversary outputs $l^{'}_{ID^*}$ pairs $\{(\sigma_{12}, z_1), (\sigma_{22}, z_2), ...(\sigma_{l^{'}_{ID^*}2}, z_{l^{'}_{ID^*}})\}$, such

that for all $1 \leq t \leq l'_{ID^*}$, $\sigma_{t2} = z_t^{\gamma}$ and number of queries to helper oracle H_O is $l_{ID^*} < l'_{ID^*}$. Thus chosen-target CDH assumption is broken.
Now Pr[Case 2b occurs] = $1-\theta$. Then,

$$Pr[Chosen-target\ CDH\ is\ solved] \geq \varepsilon_{adv}.(1-\tau).(\frac{1}{q_{H_1}}).(1-\theta) \tag{6}$$

From (4),(5) and (6),

$$\begin{aligned} Pr[Hard\ problem\ instance\ is\ solved] &\geq \varepsilon_{adv}.\tau.(1-\tfrac{1}{q}) + \varepsilon_{adv}.(1-\tau).(\tfrac{1}{q_{H_1}}).\theta + \varepsilon_{adv}.(1-\tau).(\tfrac{1}{q_{H_1}}).(1-\theta) \\ &\geq \varepsilon_{adv}.\tau.(1-\tfrac{1}{q}) + \varepsilon_{adv}.(1-\tau).(\tfrac{1}{q_{H_1}}) \end{aligned}$$

which is non-negligible for $0 \leq \tau \leq 1$ because q_{H_1} is polynomial in terms of security parameter κ. Therefore, if a Type I adversary $\mathscr{A}_I$ who can break our scheme exists, then either CDH or chosen-target CDH problem is solved.

5.1.2 Security Analysis Against Type II Adversary

Lemma 2. *If there exists a Type II adversary $\mathscr{A}_{II}$ that succeeds in winning (l, l') strong unforgeability under chosen message attack with non-negligible probability in polynomial time in random oracle model, then the chosen-target CDH problem can be solved with non-negligible probability in polynomial time.*

Proof:- Let $\mathscr{A}_{II}$ be a Type II adversary who has an advantage ε_{adv} in breaking the unforgeability property of the proposed scheme. We want to build a simulated algorithm using $\mathscr{A}_{II}$ which solves chosen-target CDH problem. Let the given chosen-target CDH instance be (g, g^{γ}) where $\gamma \in_R \mathbb{Z}_q^*$. A target oracle T_O outputting random elements z_i in $\mathbb{G}$ and helper oracle $H_O = (.)^{\gamma}$ is provided. There are different oracles similar to Type I adversary except *ReplacePK* oracle. Also we assume that each oracle maintains a list of answered queries for consistent replies to the same query. Let these list be called $L_{H_1}, L_{H_2}, L_{PartSK}$ and L_{PK}.

We set $P_{pub} = g^x$ where $x \in_R \mathbb{Z}_q^*$ and params=$(e, \mathbb{G}, \mathbb{G}_1, q, g, P_{pub}, H_1, H_2)$. Master secret key $MSK = x$ and params are given to $\mathscr{A}_{II}$. Let the maximum number of *RequestPK* queries allowed be q_{RPK}. We choose $j \in_R \mathbb{Z}_q^*$ such that $1 \leq j \leq q_{RPK}$. Simulation game will be as follows.

- H_1 **Oracle**$(ID_i||\beta)$: $\mathscr{A}_{II}$ queries challenger for H_1 value by inputting ID_i and β. Challenger will output the value as follows. If the H_1 value corresponding to the identity ID_i is already exists in the L_{H_1} list as $< ID_i, \beta, H_1(ID_i||\beta) >$, return $H_1(ID_i||\beta)$. Else, choose $r_i \in_R \mathbb{Z}_q^*$, calculate g^{r_i} and assign $H_1(ID_i||\beta) = g^{r_i}$. Append $< ID_i, \beta, H_1(ID_i||\beta) >$ to the L_{H_1} list. Return $H_1(ID_i||\beta)$ to $\mathscr{A}_{II}$.

- H_2 **Oracle**(m_i, PK_i): On inputting m_i and PK_i, if there exists an entry $< m_i, PK_i, z_i >$ in L_{H_2} list, return z_i. Else, if $PK_i = g^{\gamma}$, get a random element $z_i \in \mathbb{G}$ from the target oracle T_O, return z_i and append entry $< m_i, PK_i, z_i >$ to list L_{H_2}. If $PK_i \neq g^{\gamma}$, get a random element $r_i' \in \mathbb{Z}_q^*$, calculate $g^{r_i'}$ and assign $H_2(m_i, PK_i) = g^{r_i'}$. Append entry $< m_i, PK_i, H_2(m_i, PK_i) >$ to list L_{H_2}.

- **ExtractPartSK Oracle**(ID_i): If the corresponding entry $< ID_i, cert_i, g^{sk_i}, sk_i >$ exists in list L_{PartSK}, return $D_i = (cert_i, g^{sk_i}, sk_i)$. If the entry does not exist, choose $sk_i \in_R \mathbb{Z}_q^*$ and compute g^{sk_i}. Compute $cert_i = [H_1(ID_i||g^{sk_i})]^x$($MSK$ is known). Append $< ID_i, D_i >$ to list L_{PartSK} and return $D_i = (cert_i, g^{sk_i}, sk_i)$.

- **RequestPK Oracle(ID_i)**: If the entry is in the list L_{PK} return PK_i. Else, if $i = j$ (we let $ID_i = ID^*$), set $PK_i = PK^* = g^{\gamma}$. Append $< ID^*, -, PK^* >$ to list L_{PK}. Else, if $i \neq j$, choose $\alpha_i \in_R \mathbb{Z}_q^*$ and output g^{α_i} as PK_i. Append entry $< ID_i, \alpha_i, PK_i >$ to list L_{PK}.

- **ExtractFullSK Oracle(ID_i)**: If $ID_i = ID^*$, abort. Else get the corresponding entries by calling $ExtractPartSK(ID_i)$ and $RequestPK(ID_i)$. Output the full secret key as $SK_i = (\alpha_i, D_i)$.

- **Issue** Oracle(m', ID_i):

  - If $ID_i \neq ID^*$, get corresponding entry $< ID_i, cert_i, g^{sk_i}, sk_i >$ from L_{PartSK}. Calculate:
    * $\sigma_2^{'} = (m^{'})^{sk_i}$
    * $\sigma_3^{'} = cert_i$
    * $\sigma_4^{'} = g^{sk_i}$

    Get corresponding entry $< ID_i, \alpha_i, PK_i >$ from L_{PK}. Calculate $\sigma_1^{'} = (m^{'})^{\alpha_i}$. Then challenger outputs the tuple $\sigma^{'} = (\sigma_1^{'}, \sigma_2^{'}, \sigma_3^{'}, \sigma_4^{'})$ and gives to $\mathscr{A}_{II}$.
  - If $ID_i = ID^*$, get corresponding entry $< ID_i, cert_i, g^{sk_i}, sk_i >$ from L_{PartSK}. Then calculate:
    * $\sigma_2^{'} = (m^{'})^{sk_i}$
    * $\sigma_3^{'} = cert_i$
    * $\sigma_4^{'} = g^{sk_i}$

    Calculate $\sigma_1^{'} = (m^{'})^{\gamma}$ by querying helper oracle H_O.
    Thus challenger outputs the tuple $\sigma^{'} = (\sigma_1^{'}, \sigma_2^{'}, \sigma_3^{'}, \sigma_4^{'})$ and gives to $\mathscr{A}_{II}$.

We can see that this simulation game is indistinguishable from the actual certificateless blind signature scheme for the view of adversary $\mathscr{A}_{II}$. Eventually, $\mathscr{A}_{II}$ engages in l successful runs of Issue oracle with the challenger and outputs $l^{'}$ tuples of (m_i, ID_i, σ_i) which are distinct and $l < l^{'}$. Now, probability that a valid signature is output for an ID for which RequestPK(ID_i) was not queried is negligible, ε_{neg}. Indeed, for such an ID_i, the public key PK_{ID_i} will not even be defined in the simulator. Hence, with probability $1 - \varepsilon_{neg}$, for all ID_i appearing in output of $\mathscr{A}_{II}$, there will exist corresponding queries to RequestPK(ID_i). Hence, the upper bound on number of IDs appearing in adversary's output is q_{RPK}.
Here also, we have $l < l^{'}$ and hence for at least one output we have $l_{ID_i} < l^{'}_{ID_i}$ where $l^{'}_{ID_i}$ is the number of valid output signatures with respect to ID_i and l_{ID_i} is the number of times adversary engaged in Issue protocol with respect to ID_i. Therefore, the probability that ID^* has this property is at least $\frac{1}{q_{RPK}}$ because adversary gains no information about the chosen ID^* and it is equally likely for each ID_i to have this property.
In case ID^* satisfies the above property, it means l_{ID^*} Issue protocol were successfully completed with respect to ID^* and hence helper oracle H_O was queried l_{ID^*} times. Let the $l^{'}_{ID^*}$ output tuples containing ID^* be (m_t, ID^*, σ_t) where $1 \leq t \leq l^{'}_{ID^*}$ and $\sigma_t = (\sigma_{t1}, \sigma_{t2}, \sigma_{t3}, \sigma_{t4})$.
Applying the verification algorithm(1), we have

$$\begin{aligned} & e(\sigma_{t1}, g) = e(H_2(m_t, PK^*), PK^*) \\ \implies & e(\sigma_{t1}, g) = e(z_t, g^{\gamma}) \\ \implies & \sigma_{t1} = (z_t)^{\gamma} \end{aligned}$$

This holds for all $1 \leq t \leq l^{'}_{ID^*}$. So adversary outputs $l^{'}_{ID^*}$ pairs $\{(\sigma_{11}, z_1), (\sigma_{21}, z_2), ...(\sigma_{l^{'}_{ID^*}1}, z_{l^{'}_{ID^*}})\}$, such that for all $1 \leq t \leq l^{'}_{ID^*}$, $\sigma_{t1} = z_t^{\gamma}$ and number of queries to helper oracle H_O is $l_{ID^*} < l^{'}_{ID^*}$. Thus chosen-target CDH assumption is broken.
Assuming Type II adversary $\mathscr{A}_{II}$, who has an advantage ε_{adv} in breaking the unforgeability property, probability of hard problem (chosen-target CDH assumption) instance is solved

$$Pr[Hard\ problem\ instance\ is\ solved] \geq \varepsilon_{adv}.(1-\varepsilon_{neg}).(\frac{1}{q_{RPK}}) \quad (7)$$

Therefore, if a Type II adversary $\mathscr{A}_{II}$ who can break the proposed scheme exists, then chosen-target CDH problem is solved.

5.2 Blindness

Theorem 2. *The proposed certificateless blind signature scheme $\Pi_{\mathscr{CLBS}}$ satisfies blindness property in random oracle model under CDH assumption, if there does not exists a probabilistic polynomial time adversary $\mathscr{A}_{Blind}$ that can win the blindness game with non-negligible probability.*

Proof: In this game, the adversary $\mathscr{A}_{Blind}$ will be the signer and challenger will be the user. Let $\mathscr{A}_{Blind}$ be an adversary who is trying to break the computational blindness property of the proposed certificateless blind signature scheme $\Pi_{\mathscr{CLBS}}$ has an advantage ε_{adv}. We build the simulation in such a way that if $\mathscr{A}_{Blind}$ can break the blindness by distinguishing the signatures σ_0, σ_1 corresponds to m_0 and m_1, then challenger is able to solve CDH problem which is hard to occur. As in the definition, given CDH instance be (g, g^a, g^b), our goal is to compute g^{ab}, where $g \in \mathbb{G}$ and $a, b \in_R \mathbb{Z}_q^*$. There are H_1 oracle, H_2 oracle and ExtractPartSK oracle. Each oracle maintains a list of answered queries so that same queries can be replied with the same answer. Let L_{H_1}, L_{H_2} and L_{PartSK} are the lists.
Challenger sets $P_{pub} = g^a$ and params=$(e, \mathbb{G}, \mathbb{G}_1, q, g, P_{pub}, H_1, H_2)$.

- H_1 **Oracle**($ID_i||\beta$): For H_1 query from adversary, challenger $\mathscr{C}$ search for the entry corresponding to ID_i in L_{PartSK} $< ID_i, cert_i, g^{sk_i}, sk_i >$ and obtains g^{sk_i}.

  1. If the entry exists,
     if $\beta = g^{sk_i}$, return g^{r_i}, where $r_i \in_R \mathbb{Z}_q^*$.
     else, return $g^{r_i b}$, where $r_i \in_R \mathbb{Z}_q^*$.
  2. If the entry doesn't exist, choose $sk_i \in_R \mathbb{Z}_q^*$ and compute g^{sk_i}.
     if $\beta = g^{sk_i}$, return g^{r_i}, where $r_i \in_R \mathbb{Z}_q^*$ and append the entry $< ID_i, g^{r_i a}, g^{sk_i}, sk_i >$ to L_{PartSK}.
     else, return $g^{r_i b}$, where $r_i \in_R \mathbb{Z}_q^*$ and append the entry $< ID_i, g^{r_i a}, g^{sk_i}, sk_i >$ to L_{PartSK}.

  Append the entry $< ID_i, \beta, H_1(ID_i||\beta) >$ to L_{H_1}.

- H_2 **Oracle**(m_i, PK_i): If there exists an entry $< m_i, PK_i, z_i >$ in L_{H_2} list, return z_i. Else, choose a random element $z_i \in \mathbb{G}$, return z_i and append entry $< m_i, PK_i, z_i >$ to list L_{H_2}.

- **ExtractPartSK Oracle**(ID_i): If the corresponding entry $< ID_i, cert_i, g^{sk_i}, sk_i >$ exists in list L_{PartSK}, return $D_i = (cert_i, g^{sk_i}, sk_i)$. If the entry does not exist, choose $sk_i \in_R \mathbb{Z}_q^*$ and compute g^{sk_i}. Compute $cert_i = [H_1(ID_i||g^{sk_i})]^a = (g^{r_i})^a = (g^a)^{r_i} = P_{pub}^{r_i}$. Append $< ID_i, D_i >$ to list L_{PartSk} and return $D_i = (cert_i, g^{sk_i}, sk_i)$.

Eventually $\mathscr{A}_{Blind}$ will output some challenge identity ID^* and two equal length messages m_0 and m_1. $\mathscr{A}_{Blind}$ will ask for the partial private key D_{ID^*} and challenger can give back $D_{ID^*} = (cert^*, g^{sk^*}, sk^*)$ where $cert^* = [H_1(ID^*||g^{sk^*})]^a = (g^{r^*})^a = (g^a)^{r^*} = (P_{pub})^{r^*}$. There is no other value is required for $\mathscr{A}_{Blind}$ as all other components required for generating full secret key is under his control. $\mathscr{A}_{Blind}$ will give m_0 and m_1 along with identity ID^* to the challenger $\mathscr{C}$.
Now $\mathscr{C}$ will randomly choose a bit $b \in_R \{0,1\}$. Then $\mathscr{A}_{Blind}$ and $\mathscr{C}$ will engage in the execution

of the two instances of the interactive signing protocol say, Issue$_{\mathscr{CLBS}}$(params,m_b, ID^*, SK_{ID^*}) and Issue$_{\mathscr{CLBS}}$(params, m_{1-b}, ID^*, SK_{ID^*}). When the execution of both instances completed $\mathscr{A}_{Blind}$ will get the tuple $(\sigma_b^*, \sigma_{1-b}^*)$. Let $\sigma_b^* = (\sigma_{b1}^*, \sigma_{b2}^*, \sigma_{b3}^*, \sigma_{b4}^*)$ and $\sigma_{1-b}^* = (\sigma_{(1-b)1}^*, \sigma_{(1-b)2}^*, \sigma_{(1-b)3}^*, \sigma_{(1-b)4}^*)$.
Now, we analyse the following 2 cases.

Case 1: Both the tuples $(\sigma_{b3}^*, \sigma_{b4}^*)$ and $(\sigma_{(1-b)3}^*, \sigma_{(1-b)4}^*)$ are equal to the tuple $(cert^*, g^{sk^*})$. In this case these two components of the signatures are of no use to the adversary $\mathscr{A}_{Blind}$ in distinguishing between b and $1-b$ as they are same for both executions. But the first two components $(\sigma_{b1}^*, \sigma_{b2}^*)$ and $(\sigma_{(1-b)1}^*, \sigma_{(1-b)2}^*)$ can cause impact on distinguishing between b and $1-b$.
As in the definition, blindness property indicates that during signing process, signer should not learn anything about the message which he signs for a particular user. User blinds the message by using randomness $r_{Blind} \in_R \mathbb{Z}_q^*$. Now we can have a claim as follows.

Claim 1: *Given valid message/signature pair components (m, σ_1, σ_2) and blind message/blind signature pair components $(m', \sigma_1', \sigma_2')$ (in the view of $\mathscr{A}_{Blind}$ during signing phase), there always exists a unique blinding factor $r_{Blind} \in_R \mathbb{Z}_q^*$ such that both the tuples have the same relation as defined by the Issue$_{\mathscr{CLBS}}$ protocol.*
We can define r_{Blind} such that $m' = H_2(m, PK_A)\ g^{r_{Blind}}$. Let $\sigma_{b4}^* = \sigma_{(1-b)4}^* = \sigma_4'$. In order to maintain the same relation in the protocol, define $\bar{\sigma}_1 = \sigma_1'(PK_A)^{-r_{Blind}}$ and $\bar{\sigma}_2 = \sigma_2'(\sigma_4')^{-r_{Blind}}$. We can now show that $(m, \bar{\sigma}_1, \bar{\sigma}_2)$ is a valid signature component and satisfies the verification conditions and thus $\sigma_1 = \bar{\sigma}_1$ and $\sigma_2 = \bar{\sigma}_2$.
Consider the first verification algorithm(1).

$$\begin{aligned}
e(\bar{\sigma}_1, g) &= e(\sigma_1'(PK_A)^{-r_{Blind}}, g) \\
&= e((m')^{\alpha_A}.(PK_A)^{-r_{Blind}}, g) \\
&= e((H_2(m, PK_A).g^{r_{Blind}})^{\alpha_A}.(PK_A)^{-r_{Blind}}, g) \\
&= e((H_2(m, PK_A))^{\alpha_A}.(g^{r_{Blind}})^{\alpha_A}.(PK_A)^{-r_{Blind}}, g) \\
&= e((H_2(m, PK_A))^{\alpha_A}.(g^{\alpha_A})^{r_{Blind}}.(g^{\alpha_A})^{-r_{Blind}}, g) \\
&= e((H_2(m, PK_A))^{\alpha_A}, g) \\
&= e(H_2(m, PK_A), g^{\alpha_A}) \\
&= e(H_2(m, PK_A), PK_A)
\end{aligned}$$

Consider the second verification algorithm (2).

$$\begin{aligned}
e(\bar{\sigma}_2, g) &= e(\sigma_2'(\sigma_4')^{-r_{Blind}}, g) \\
&= e((m')^{sk_A}.(\sigma_4')^{-r_{Blind}}, g) \\
&= e((H_2(m, PK_A).g^{r_{Blind}})^{sk_A}.(\sigma_4')^{-r_{Blind}}, g) \\
&= e((H_2(m, PK_A))^{sk_A}.(g^{r_{Blind}})^{sk_A}.(\sigma_4')^{-r_{Blind}}, g) \\
&= e((H_2(m, PK_A))^{sk_A}.(g^{sk_A})^{r_{Blind}}.(g^{sk_A})^{-r_{Blind}}, g) \\
&= e((H_2(m, PK_A))^{sk_A}, g) \\
&= e(H_2(m, PK_A), g^{sk_A}) \\
&= e(H_2(m, PK_A), \sigma_4)
\end{aligned}$$

We can see that verification conditions (1) and (2) are satisfied, therefore our claim is valid.
Due to *Claim 1*, there always exists a random blinding factor $r_{Blind} \in_R \mathbb{Z}_q^*$ connecting a valid blind signature to any view of the adversary $\mathscr{A}_{Blind}$. Since $(\sigma_{b3}^*, \sigma_{b4}^*) = (\sigma_{(1-b)3}^*, \sigma_{(1-b)4}^*)$ and the view of $\mathscr{A}_{Blind}$ while

interacting with user U is equally likely to be connected to the signature $\sigma(m_b, ID^*)$ and the signature $\sigma(m_{1-b}, ID^*)$. Thus the signatures generated $(\sigma_b^*, \sigma_{1-b}^*)$ by interacting $\mathscr{A}_{Blind}$ with challenger with two instances of Issue$_{\mathscr{CLBS}}$(params,m_b, ID^*, SK_{ID^*}) and Issue$_{\mathscr{CLBS}}$ (params,m_{1-b}, ID^*, SK_{ID^*}) are equally likely. Therefore, the probability of an adversary $\mathscr{A}_{Blind}$ outputs the correct value $b^{'}$ is equal to $\frac{1}{2}$ which is the guessing probability. Thus $\mathscr{A}_{Blind}$ cannot gain any advantage to distinguish between two messages m_0 and m_1 during the signing process. We define the success probability that $\mathscr{A}_{Blind}$ wins in case 1 as Pr[$\mathscr{A}_{Blind}$ outputs $(b = b^{'})$] $= \frac{1}{2}$.

Case 2: At least one of the tuples $(\sigma_{b3}^*, \sigma_{b4}^*)$ and $(\sigma_{(1-b)3}^*, \sigma_{(1-b)4}^*)$ are not equal to the tuple $(cert^*, g^{sk^*})$. Let that tuple be $(\sigma_{j3}^*, \sigma_{j4}^*)$ where $j = b$ or $j = 1-b$. Since $\sigma_{j4}^* \neq g^{sk^*}$ by definition of H_1 oracle, $H_1(ID^*||\sigma_{j4}^*) = g^{r^*b}$. If $(\sigma_{j3}^*, \sigma_{j4}^*)$ is valid signature components, it will satisfy the verification algorithm (3) as follows.

$$
\begin{aligned}
e(\sigma_{j3}^*, g) &= e((H_1(ID^*||\sigma_{j4}^*)), P_{pub}) \\
e(\sigma_{j3}^*, g) &= e((H_1(ID^*||\sigma_{j4}^*)), g^a) \\
e(\sigma_{j3}^*, g) &= e(g^{r^*b}, g^a) \\
\sigma_{j3} &= g^{r^*ab}
\end{aligned}
$$

r^* is chosen by the challenger during H_1 oracle query phase and so challenger can compute $(\sigma_{j3})^{\frac{1}{r^*}} = (g^{r^*ab})^{\frac{1}{r^*}} = g^{ab}$. Therefore, case 2 is equivalent to solve CDH problem.
Assuming adversary $\mathscr{A}_{Blind}$, who has an advantage ε_{adv} in breaking the blindness property. Let $\sigma_j^* = (\sigma_{j1}^*, \sigma_{j2}^*, \sigma_{j3}^*, \sigma_{j4}^*)$ where $j = b$ or $j = 1-b$. sk^* is chosen completely random during simulation and g^{sk^*} is independent of any other parameters or protocols. Hence, adversary can know about g^{sk^*} only with negligible probability. That is,

$\quad$ Pr[$\sigma_{j4}^* = g^{sk^*}$]=$\frac{1}{q}$
$\implies$ Pr[$\sigma_{j4}^* \neq g^{sk^*}$]=$(1-\frac{1}{q})$ [Acco. to case 2: $\sigma_{j4}^* \neq g^{sk^*}$]
$\implies$ Pr[$H_1(ID^*||\sigma_{j4}^*) = g^{r^*b}$]=$(1-\frac{1}{q})$ [By definition of H_1 oracle]

Let Pr[Case 2 occurs] = δ. Assuming $\mathscr{A}_{Blind}$, who has an advantage ε_{adv} in breaking the blindness property, then,

$$
Pr[Hard\ problem(CDH)\ instance\ is\ solved] \geq \varepsilon_{adv}.\delta.(1-\frac{1}{q}) \tag{8}
$$

5.3 Comparison

Different certificateless blind signatures($\mathscr{CLBS}$) are available in the literature [19, 20, 21, 22], but the existing schemes lack explicit mathematical security proofs. As shown in Fig. 1, compared to existing schemes, our scheme is the only one which is proven to be strongly unforgeable against $(l, l^{'})$ adversary. Even though our scheme has more signature components, it doesnot cause computational overhead because two signature components(σ_3, σ_4) are precomputed values. Also our scheme is the only scheme with probability analysis.

6 Conclusion

We designed a concrete construction of a certificateless blind signature scheme, with formal proof of security for both unforgeability and blindness. The proof of security of the proposed scheme is presented in the random oracle model. Our scheme could provide rigorous formal proof without causing additional

Scheme	Security Proof	Hardness Assumption	Probability(Advantage) of the Adversary
Yang et al.[19]	Existentially Unforgeable (Only Theorem statement)	Type-1 k-CCA Type-2 mICDH	No Probability Analysis is given
Sun et al. Scheme 1 [20]	Unforgeable (similar to Choi et al.'s [22]CLS scheme)	Type-1 CDH Type-2 mICDH	No Probability Analysis is given
Sun et al. Scheme 2 [20]	Unforgeable (similar to Choi et al.'s [22]CLS scheme)	Type-1 k-CAA Type-2 mICDH	No Probability Analysis is given
Zhang et al.[21]	Unforgeable (Only Theorem statement)	Type-1 q-SDH Type-2 BPI	No Probability Analysis is given
Our Scheme	Strongly Unforgeable (Both Type 1 and 2) Detailed Proof is given	Type-1 CDH and ct-CDH Type-2 ct-CDH Blindness CDH	Detailed Probability Analysis is given

Figure 1: Comparison with existing schemes

computaional overhead. As per our knowledge there is no certificateless blind signature in the standard model. A scheme with standard model proof can be a good problem to work on. Since there are efficient signature schemes like Water's signature[23] in the standard model, we can easily make it as a certificateless blind signature. But rigorous work has to be done for designing the security proof in standard model since there are different types of adversaries in certificateless scenario and also need to prove both unforgeability and blindness.

References

[1] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Proc. of the 9th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'03), Taipei, Taiwan, LNCS*, vol. 2894. Springer-Verlag, November 2003, pp. 452–473.

[2] Z. Zhang, D. S. Wong, J. Xu, and D. Feng, "Certificateless public-key signature: Security model and efficient construction," in *Proc. of the 4th International Conference on Applied Cryptography and Network Security (ACNS'06), Singapore, LNCS*, vol. 3989. Springer-Verlag, June 2006, pp. 293–308.

[3] J. K. Liu, M. H. Au, and W. Susilo, "Self-generated-certificate public key cryptography and certificateless signature / encryption scheme in the standard model: extended abstract," in *Proc. of the 2007 ACM Symposium on Information, Computer and Communications Security (ASIACCS'07), Singapore*. ACM, March 2007, pp. 273–283.

[4] X. Huang, Y. Mu, W. Susilo, D. S. Wong, and W. Wu, "Certificateless signature revisited," in *Proc. of the 12th Australasian Conference on Information Security and Privacy (ACISP'07), Townsville, Australia, LNCS*, vol. 4586. Springer-Verlag, July 2007, pp. 308–322.

[5] D. Chaum, "Blind signatures for untraceable payments," in *Proc. of the 2nd Annual International Conference on Advances in Cryptology (CRYPTO'82), Santa Barbara, California, USA*. Springer-Verlag, August 1983, pp. 199–203.

[6] D. Schröder and D. Unruh, "Security of blind signatures revisited," in *Proc. of the 15th International Conference on Practice and Theory in Public Key Cryptography (PKC'12), Darmstadt, Germany, LNCS*, vol. 7293. Springer-Verlag, May 2012, pp. 662–679.

[7] D. Pointcheval and J. Stern, "Provably secure blind signature schemes," in *Proc. of the 1996 International Conference on the Theory and Applications of Cryptology and Information Security (ASIACRYPT'96), Kyongju, Korea, LNCS*, vol. 1163. Springer-Verlag, November 1996, pp. 252–265.

[8] A. Juels, M. Luby, and R. Ostrovsky, "Security of blind digital signatures (extended abstract)," in *Proc. of the 17th Annual International Conference on Advances in Cryptology (CRYPTO'97), Santa Barbara, California, USA, LNCS*, vol. 1294. Springer-Verlag, August 1997, pp. 150–164.

[9] A. Boldyreva, “”threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme”,” in *Proc. of the 6th International Workshop on Theory and Practice in Public Key Cryptography (PKC'03), Miami, Florida, USA, LNCS*, vol. 2567, January 2003, pp. 31–46.

[10] M. Fischlin, “Round-optimal composable blind signatures in the common reference string model,” in *Proc. of the 26th Annual International Conference on Advances in Cryptology (CRYPTO'06), Santa Barbara, California, USA, LNCS*, vol. 4117. Springer-Verlag, August 2006, pp. 60–77.

[11] C. Hazay, J. Katz, C.-Y. Koo, and Y. Lindell, “Concurrently-secure blind signatures without random oracles or setup assumptions,” in *Proc. of the 4th Theory of Cryptography Conference (TCC'07), Amsterdam, The Netherlands, LNCS*, vol. 4392. Springer-Verlag, February 2007, pp. 323–341.

[12] T. Okamoto, “Efficient blind and partially blind signatures without random oracles,” in *Proc. of the 3rd Theory of Cryptography Conference (TCC'06), New York, NY, USA*, vol. 3876. Springer-Verlag, March 2006, pp. 80–99.

[13] J. Camenisch, M. Koprowski, and B. Warinschi, “Efficient blind signatures without random oracles,” in *Proc. of the 4th International Conference of Security in Communication Networks (SCN'04), Amalfi, Italy, LNCS*, vol. 3352. Springer-Verlag, September 2004, pp. 134–148.

[14] E. Ghadafi and N. P. Smart, “Efficient two-move blind signatures in the common reference string model,” in *Proc. of the 15th International Conference on Information Security (ISC'12), Passau, Germany, LNCS*, vol. 7483. Springer-Verlag, September 2012, pp. 274–289.

[15] F. Zhang and K. Kim, “ID-based blind signature and ring signature from pairings,” in *Proc. of the 8th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'02), Queenstown, New Zealand, LNCS*, vol. 2501. Springer-Verlag, December 2002, pp. 533–547.

[16] F. Zhang and K. Kim, “Efficient ID-based blind signature and proxy signature,” in *Proc. of the 8th Australasian Conference on Information Security and Privacy (ACISP'03), Wollongong, Australia, LNCS*, vol. 2727. Springer-Verlag, July 2003, pp. 312–323.

[17] D. Galindo, J. Herranz, and E. Kiltz, “On the generic construction of identity-based signatures with additional properties,” in *Proc. of the 12th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'06), Shanghai, China, LNCS*, vol. 4284. Springer-Verlag, December 2006, pp. 178–193.

[18] L. Zhang and F. Zhang, “Certificateless signature and blind signature,” *Journal of Electronics (China)*, vol. 25, no. 5, pp. 629–635, September 2008.

[19] X. Yang, Z. Liang, P. Wei, and J. Shen, “A provably secure certificateless blind signature scheme,” in *Proc. of the 5th International Conference on Information Assurance and Security (IAS'09), Xi'an, China*, vol. 2. IEEE, August 2009, pp. 643–646.

[20] S. Sun and Q. Wen, “Novel efficient certificateless blind signature schemes,” in *Proc. of the 2009 International Symposium on Computer Network and Multimedia Technology (CNMT'09), Wuhan, China*. IEEE, January 2009, pp. 1–5.

[21] J. Zhang and S. Gao, “Efficient provable certificateless blind signature scheme,” in *Proc. of the 2010 International Conference on Networking, Sensing and Control (ICNSC'10), Chicago, Illinois, USA*. IEEE, April 2010, pp. 292 –297.

[22] H. Y. Choi, J. H. Park, J. Y. Hwang, and D. H. Lee, “Efficient certificateless signature schemes,” in *Proc. of the 5th International Conference on Applied Cryptography and Network Security (ACNS'07), Zhuhai, China, LNCS*, vol. 4521. Springer-Verlag, June 2007, pp. 443–458.

[23] B. Waters, “Efficient identity-based encryption without random oracles,” in *Proc. of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques (Eurocrypt'05), Aarhus, Denmark, LNCS*, vol. 3494. Springer-Verlag, May 2005, pp. 114–127.

Author Biography

Sangeetha Jose is a Ph D scholar from Theoretical Computer Science Lab at Indian Institute of Technology (IIT) Madras, Chennai, India. She is working under the guidance of Prof. C. Pandu Rangan. Her research interests are in provable security mainly focus on the design and analysis of public key encryption and digital signatures and the security of cloud computing. Contact her at sangeethajosem@gmail.com.

Akash Gautam[1] was a dual degree student in Indian Institute of Technology (IIT) Madras, Chennai, India. He was working under the guidance of Prof C. Pandu Rangan. His areas of interest focus on provably secure post quantum cryptosystems. Akash Gautam can be contacted at akash.gautam24@gmail.com.

C. Pandu Rangan is Professor in the department of computer science and engineering of Indian Institute of Technology (IIT) Madras, Chennai, India. Theoretical Computer Science Lab in IIT Madras is headed by him. His areas of research interests mainly focus on cryptography, algorithms and data structures, game theory, graph theory and distributed computing. C. Pandu Rangan can be contacted at prangan55@gmail.com

[1]No photo is available.