# Strongly Secure Password Based Blind Signature for Real World Applications

Sangeetha Jose, Preetha Mathew K. , C. Pandu Rangan

*Abstract*—**Digital signature is the cryptographic primitive that ensures authentication and nonrepudiation. A password based blind signature can be used in the scenarios, where the participation of both the signer and the user are required. The user requires the authentication of the signer without revealing the message to the signer. This requirement is needed for real world applications such as client server applications in the banking scenario. As per our knowledge, the first password based blind short signature was constructed by Sangeetha et al. in CECC 2013 which ensures the properties unforgeability, blindness and unframeability. But if the password size is very small, it may be susceptible to off-line password guessing attack. In this paper we propose a strongly secure password based blind short signature which solves the off-line password guessing attack. The formal proof of the scheme is reduced to computational Diffie-Hellman(CDH) assumption.**

*Index Terms*—**Blind Signature Scheme, Password Based Blind Signature Scheme, Unforgeability, Blindness, Unframeability.**

## I. Introduction

Eventhough there are tremendous growth in technologies in this twenty first century, secure data transmission is still appear to be a big hurdle and a lot of security issues need to be solved. Encryption schemes provide confidentiality where as digital signatures provide unforgeability. Digital signature scheme allows to sign documents in such a way that anyone can verify the authenticity of the signature. Diffie and Hellman [6] coined the notion of public key cryptosystem and Rivest et al. [7] proposed the first known digital signature called RSA(Rivest, Shamir and Adleman) signature scheme. The definition of security requirements for signature scheme was given by Goldwasser et al. [11] and the security proof for signature scheme in random oracle model was proposed by Pointchevel et al. [13]. The cryptosystems which is proved to be secure with random oracle uses cryptographic hash functions(preimage and collision resistant) and in the proof of security we assume that the output of hash functions follows a uniform distribution. Bellare et al. also had given the security proof of a RSA based digital signature in their classical work [1].

The idea of blind signature was put forwarded by David Chaum [5]. The applications like e-voting, digital cash etc require signatures which conceal the original message. 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 [8]. The properties of blind signatures are blindness and unforgeability. The provable secure design for blind signature is proposed by Pointchevel et al. [12] in which they defined the security for blind signatures with an application to electronic cash. Security arguments for blind signatures are proposed in papers([10],[14],[8]).

Gjosteen et al. [9] presented password based signature schemes based on RSA(Rivest, Shamir and Adleman) assumption and LRSW (Lysyanskaya, Rivest, Sahai and Wolf) assumption in which password is used as a random seed for the digital signature's key generation algorithm. Since passwords are short compared to key size, the key storage constraints can be solved. But these kind of schemes may be susceptible to online and off-line password guessing attacks for the low entropy passwords. In cryptography, Shannon([2],[3]) coined the term "entropy" which has been used as a measure of the difficulty in guessing or finding a password or a key. According to the NIST(National Institute of Standards and Technology) recommendations [4], 80 bits entropy are required for secure passwords. But passwords should be randomly selected passwords. Then the minimum threshold level of entropy can be obtained by using minimum 13 characters for the password from a 94 printable characters (Entropy, $H = log_2(b^l) \approx 85$ bits, where $b = 94$ and $l = 13$) which ensures the secrecy of the passwords. In different banking applications like e-locker facility, the secret information of the customer and the bank are together needed for transaction. For this purpose it is essential to generate signature mutually by using both secret key of customer and banker's secret key. For signature generation if customer is using certain threshold passwords along with banker's secret key it will increase the security as well as the efficiency of the system because customers can remember comparatively smaller passwords rather than using a large secret key. This insight motivates the construction of the password based blind signature(PBBS) scheme described in [21] in which both user's password and server's secret key are simultaneously used for signature generation.

**Related Work:** Gjosteen et al. [9] proposed password based signatures which prevents dictionary attacks. They introduced two password based signature schemes based on RSA [7] and CL(Camenisch and Lysyanskaya) [15] signatures. First scheme is easy to implement, but it does not achieve the security requirements. Second scheme is less practical, but it achieves stronger security. Password based signatures have a lot of applications in the banking scenario. Hence a password based

Sangeetha Jose is a Ph D scholar from Theoretical Computer Science Lab at Indian Institute of Technology Madras. (email: sangeethajosem@gmail.com).

Preetha Mathew K. is a Ph D scholar from Theoretical Computer Science Lab at Indian Institute of Technology Madras. (email: preetha.mathew.k@gmail.com).

C. Pandu Rangan is Professor in Department of Computer Science and Engineering at Indian Institute of Technology Madras(email: prangan55@gmail.com).

blind signature(PBBS) scheme is proposed in [21] by making use of blind version of BLS(Boneh, Lynn and Shacham) short signature scheme([17],[18]). In this paper we modified [21] to obtain a strongly secure password based blind signature(ss-PBBS).

**Motivation**: In all client server environment applications, if we use client's password as well as server's secret key for signing a document so as to ensure high efficiency and security. That is, client(user) and server(signer) can sign the document only by mutual agreement, so that user cannot generate signature without secret key of the signer(unforgeability) and the signer is not able to sign on behalf of the user without users password(unframeability). If the server's signature can be obtained by the client without revealing the message to the server, it is called blindness. To achieve the goals of unforgeability, unframeability and blindness, a password based blind signature construction is required which uses secret of both the client and server. This stimulates for the construction of a new password based blind signature(PBBS) scheme [21] in which message is being signed using both client's password as well as server's secret key. The password based blind signature scheme is based on blind version of BLS short signature scheme, which significantly reduces the signature size to 170 bits compared to Gjosteen et al.s password based signatures with 1024 bits and $2\kappa$ bits where $\kappa$ is security parameter which is considered to be large. This scheme can be effectively used in banking applications. The key construction of the scheme is similar to Gjosteen et al.[9], but the rest of the construction is entirely different as shown in Table 1. Security proof of the scheme is elaborately given which is based on computational Diffie-Hellman(CDH) assumption in random oracle model. But there is a constraint in the size of password. In order to overcome this drawback we designed a strongly secure password based blind signature.

### A. *Organization of the Paper*

Section 2 explains the preliminary concepts of bilinear pairing and the hardness assumptions which helps to prove the security of schemes. Section 3 gives the definitions of password based blind signatures and its security. Section 4 explains the password based blind signature scheme in [21]. Section 5 discusses strongly secure password based blind signature scheme, the proof of security and its advantages. The paper concludes in section 6.

## II. Preliminary Concepts

### A. *Bilinear Pairing*

Let $\mathbb{G}_1$ be a multiplicative cyclic prime order group $q$ with generator $g$ and $\mathbb{G}_2$ also be a multiplicative cyclic group of the same prime order $q$. A map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is said to be a bilinear pairing if the following properties hold.

1. $Bilinearity$: For all $g \in \mathbb{G}_1$ and $a, b \in_R \mathbb{Z}_q^*$, $e(g^a, g^b) = e(g,g)^{ab}$.
2. $Non\text{-}degeneracy$: For all $g \in \mathbb{G}_1$, $e(g,g) \neq I_{\mathbb{G}_2}$ where $I_{\mathbb{G}_2}$ is the identity element of $\mathbb{G}_2$.
3. $Computability$: $e$ is efficiently computable.

### B. *Computational Diffie-Hellman (CDH) Assumption*

Security proof of scheme is based on CDH assumption. CDH problem states that given $(g, g^a, g^b)$, compute $g^{ab}$, where $g \in \mathbb{G}_1$ and $a, b \in_R \mathbb{Z}_q^*$.

*Definition 1:* **(CDH Assumption):** The advantage of any probabilistic polynomial time algorithm $\mathcal{A}$ in solving the CDH problem in $\mathbb{G}_1$ is defined as
$Adv_{\mathcal{A}}^{CDH} = Prob[g^{ab} \leftarrow \mathcal{A}(g, g^a, g^b) \mid g \in \mathbb{G}_1$ and $a, b \in_R \mathbb{Z}_q^*]$
The Computational Diffie-Hellman(CDH) assumption is that, for any probabilistic polynomial time algorithm $\mathcal{A}$, the advantage $Adv_{\mathcal{A}}^{CDH}$ is negligibly small($\epsilon$).

### C. *Conference-key Sharing Scheme (CONF)*

CONF states that given $(g, g^a, g^{ab})$, compute $g^b$, where $g \in \mathbb{G}_1$ and $a, b \in_R \mathbb{Z}_q^*$.

*Definition 2:* **(CONF Assumption):** The advantage of any probabilistic polynomial time algorithm $\mathcal{A}$ in solving the CONF problem in $\mathbb{G}_1$ is defined as
$Adv_{\mathcal{A}}^{CONF} = Prob[g^{b} \leftarrow \mathcal{A}(g, g^a, g^{ab}) \mid g \in \mathbb{G}_1$ and $a, b \in_R \mathbb{Z}_q^*]$

## III. Definition of Password Based Blind Signatures

Password based blind signature consists of different algorithms which is defined as follows [9].

*Definition 3:* **(Password Based Blind Signatures):** A password based blind signature scheme mainly consists of the following six algorithms.

- Setup($1^\kappa$): A trusted third party outputs the public parameters by accepting the security parameter $\kappa$ as input. It includes group parameters, message space, password space, hash functions, mappings etc. The parameters have public access by all the algorithms.
- KeyGen: These are interactive algorithms run by user and server. This algorithm inputs user password $pw$ and outputs the values needed for obtaining signing key($sk_{PBBS}$) of the server. It also generates secret and public keys($sk$ and $pk$) of both the user and server.
- Request($m, pk, pw$): User runs this algorithm on message $m$ and outputs the signature request $L$ and the state information.
- Issue($L, pk, sk_{PBBS}$): Server runs this algorithm in which signature request $L$ is the input and the output is blind signature $\sigma'$.
- Unblind($\sigma', pk, state$): This algorithm is also run by the user. This makes use of blind signature $\sigma'$, public keys and $state$ from $Request$ algorithm and outputs signature $\sigma$. But when the check fails, algorithm outputs $\perp$.
- Verify($m, \sigma, pk$): Anyone can verify that whether $\sigma$ is a valid signature on $m$ under publicly available information like $pk$ by $Verify$ algorithm. If it is a valid signature algorithm outputs 1, otherwise outputs 0.

User has a secret password with a minimum level of entropy which is chosen randomly.

## A. Security Definitions of Password Based Blind Signatures

The security of the password based blind signatures can be convinced by proving its different properties which are as follows, *unforgeability, blindness and unframeability*. The additional property which is present in PBBS is that of $unframeability$. The other two, viz $unforgeability$ and $blindness$ are the properties of blind signature. The formal definition of the said properties are detailed below.

*1) **Unforgeability**:* In the formal definition of unforgeability, where the adversary $\mathcal{A}$ plays the role of user and the simulator will have the role of server. This game is based on random oracle and the challenger has to provide hash oracle and sign oracle($Issue$) and $\mathcal{A}$ tries to get "one-more" signature.

*Definition 4:* **(Unforgeability) [10]:** A password based blind signature scheme PBBS is said to be unforgeable, if the probability that $\mathcal{A}$ wins the following game is negligible.

- Step 1 (Setup Phase): $(pk, sk) \leftarrow KeyGen(1^{\kappa})$.
- Step 2 (Training Phase): $\mathcal{A}$ engages in polynomially many(in $\kappa$) adaptive interactive protocols (hash and Issue oracles) with polynomially many copies of server($pk, sk$). Let '$l$' be the number of executions in which server outputs valid message-signature pair at the end of step 2.
- Step 3 (Forgery Phase): $\mathcal{A}$ outputs a set of $\{(m_1, \sigma_1), ..., (m_j, \sigma_j)\}$ where $(m_i, \sigma_i)$ for $1 \leq i \leq j$ are all accepted by $Verify(m_i, pk, \sigma_i)$ for distinct $m_i$.

We can say that $\mathcal{A}$ wins the game when $j > l$. That is, $\mathcal{A}$ outputs more valid tuples $(m, \sigma)$ than he/she received during the training phase.

*2) **Blindness**:* It ensures that server cannot distinguish between two messages $m_0, m_1$ which has already signed by him with the interaction of the user. For proving blindness, server plays the role of adversary $\mathcal{A}$ and challenger $\mathcal{C}$ will be the user.

*Definition 5:* **(Blindness) [10]:** A password based blind signature scheme PBBS satisfies the property of blindness, if the probability that $\mathcal{A}$ wins the following game is negligible.

- Step 1: $(pk, sk) \leftarrow KeyGen(1^{\kappa})$
- Step 2: $\mathcal{A}$ produces two messages $\{m_0, m_1\}$ polynomial in $1^{\kappa}$ where $\{m_0, m_1\}$ are by convention lexicographically ordered and give to the $\mathcal{C}$.
- Step 3: $\{m_b, m_{1-b}\}$ are the same messages $\{m_0, m_1\}$ ordered by $\mathcal{C}$ according to the value of bit $b \in \{0, 1\}$ which is hidden from $\mathcal{A}$. $\mathcal{A}$ has given access to two interactive protocols with user $\mathcal{U}$, first with $\mathcal{U}(params, pk, m_b)$ and second with $\mathcal{U}(params, pk, m_{1-b})$.
- Step 4: Initially if the user protocol's output is $\sigma_b$(that is, does not output fail) and the next time user protocol's output is $\sigma_{1-b}$,(that is, does not output fail) then only $\mathcal{A}$ gets $\sigma_b, \sigma_{1-b}$ ordered according to the corresponding $(m_0, m_1)$.
- Step 5: $\mathcal{A}$ outputs a bit $b^{'}$.

We can see that $\mathcal{A}$ can predict $b' = b$ only with a guessing probability. Therefore, we can define adversary $\mathcal{A}$'s advantage in the game as $|Pr[b' = b] - 1/2|$. That is, the server is not able to distinguish the messages that he/she signs in the previous sessions.

*3) **Unframeability**:* This is an additional property which is required for proving the security of the password based blind signature schemes. This property ensures that the server is not able to sign on behalf of the user without user's knowledge. Otherwise server has to find out user's password. Thus server will be the adversary $\mathcal{A}$ and tries to construct password based signature without the user intervention of the user. The formal definition of unframeability is as follows.

*Definition 6:* **(Unframeability) [9]:** A password based blind signature scheme PBBS is unframeable, if the probability that $\mathcal{A}$ wins the following game is negligible.

- Step 1 (Setup Phase): $(pk, sk) \leftarrow KeyGen(1^{\kappa})$
- Step 2 (Training Phase): $\mathcal{A}$ engages in polynomially many(in $\kappa$) adaptive interactive protocols (hash, Request and Unblind oracles) with polynomially many copies of user($pk, sk$). $\mathcal{A}$ can ask any number of queries to this oracles and decides in an adaptive fashion when to stop.
- Step 3 (Frameability Phase): $\mathcal{A}$ outputs a $(m^*, \sigma^*)$ which has to be verified by $Verify(m^*, pk, \sigma^*)$ algorithm for a distinct $m^*$.

We say that $\mathcal{A}$ wins the game when Verify$(m^*, pk, \sigma^*) = 1$. That is, $\mathcal{A}$ outputs a valid tuple $(m^*, \sigma^*)$ other than he/she received during the training phase without the help of the user.

## IV. Password Based Blind Signature(PBBS)

Password based blind signature(PBBS) in [21] is shown in Fig. 1 which is an interaction between a user and a server(signer). The authentication protocol should be resistant to eavesdropping attacks, so that the protocol should not be attacked by an adversary to carry out offline attack. Here anyone can have a feel that if we expose $y = g^{H_2(pw)}$ as public key, it is susceptible to offline guessing attacks. But since the password is randomly selected, we can ensure the security by using 13 character passwords. According to the NIST(National Institute of Standards and Technology) recommendations [4], 80 bits entropy are required for secure passwords. The minimum threshold level of entropy can be obtained by using minimum 13 characters for a randomly selected password from a 94 printable characters (Entropy, $H = log_2(b^l) \approx 85$ bits, where b = 94 and l = 13) which ensures the security of the passwords. That is, it is quite infeasible for an attacker to do offline guessing in polynomial time.

**Verification** algorithm(Verify$(m, \sigma, y_2, y)$) helps to verify the validity of the message-signature pair.

```
if $e(\sigma, g) \overset{?}{=} e(H_1(m), y_2 y)$
    $return$ 1
else $return$ 0
```

To show the **correctness** of verification algorithm(Verify$(m, \sigma, y_2, y)$), the equation can be expanded as follows.

Note that $\sigma = \dfrac{\sigma' L^{H_2(pw)} H_1(m)^{\eta}}{(y_1 y_2)^k}$

$$= \frac{L^{x_2 - \eta} L^{H_2(pw)} H_1(m)^{H_2(pw) - x_1}}{(g^{x_1} g^{x_2})^k}$$

**Setup**($1^\kappa$): Select a pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ where $\mathbb{G}_1$ and $\mathbb{G}_2$ are cyclic prime order group in $q$ and a generator $g \in \mathbb{G}_1$. Select hash functions, $H_1 : \{0,1\}^* \to \mathbb{G}_1$ and $H_2 : \{0,1\}^* \to \mathbb{Z}_q{}^*$ and return public parameters $params \leftarrow (e, q, \mathbb{G}_1, \mathbb{G}_2, g, H_1, H_2)$.

```
USER                                                    SIGNER

KeyGen_U($pw$):                                         KeyGen_S($\eta$):
  $x_1 \leftarrow_R \mathbb{Z}_q^*$                                      $x_2 \leftarrow_R \mathbb{Z}_q^*$
  $y_1 \leftarrow g^{x_1}$                                        $y_2 \leftarrow g^{x_2}$
  $y \leftarrow g^{H_2(pw)}$
  $\eta \leftarrow H_2(pw) - x_1$         $\xrightarrow{\eta}$          Signing Key, $sk_{PBBS} = x_2 - \eta$
  $return(x_1, y_1, y, \eta)$                             $return(x_2, sk_{PBBS}, y_2)$

Secret Keys(sk): $sk_{\mathcal{U}} = x_1, sk_{\mathcal{S}} = x_2$,    Public Keys(pk): $pk_{\mathcal{U}} = y_1, pk_{\mathcal{S}} = y_2$, $y = g^{H_2(pw)}$

Request_U($m, pk, pw$):
  $k \leftarrow_R \mathbb{Z}_q^*$
  $L = H_1(m)g^k$                 $\xrightarrow{L}$          Issue_S($L, pk, sk_{PBBS}$):
  $state \leftarrow (m, k, pw)$                               $\sigma' = (L)^{sk_{PBBS}}$
                                  $\xleftarrow{\sigma'}$
Unblind_U($\sigma', pk, state$):
  if ($e(L, y_2 g^{-\eta}) \overset{?}{=} e(\sigma', g)$)
     then
        $\sigma = \frac{\sigma' L^{H_2(pw)} H_1(m)^\eta}{(y_1 y_2)^k}$
        if $Verify(m, \sigma, y_2, y) = 1$
           then $return(\sigma)$
  $return\ (\perp)$
```

Fig. 1. Password based blind signature scheme(PBBS) in [21]

$$\begin{aligned}
&= \frac{L^{x_2+H_2(pw)-\eta} H_1(m)^{H_2(pw)-x_1}}{(g^{x_1}g^{x_2})^k} \\
&= \frac{L^{x_1+x_2} H_1(m)^{H_2(pw)-x_1}}{g^{k(x_1+x_2)}} \\
&= \frac{(H_1(m)g^k)^{x_1+x_2} H_1(m)^{H_2(pw)-x_1}}{g^{k(x_1+x_2)}} \\
&= H_1(m)^{x_1+x_2+H_2(pw)-x_1} \\
&= H_1(m)^{x_2+H_2(pw)}
\end{aligned}$$

Therefore,

$$\begin{aligned}
e(\sigma, g) &= e(H_1(m)^{x_2+H_2(pw)}, g) \\
&= e(H_1(m), g^{x_2}g^{H_2(pw)}) \\
&= e(H_1(m), y_2 y)
\end{aligned}$$

## V. Strongly Secure Password Based Blind Signature Scheme(ss-PBBS)

The strongly secure scheme is as in Fig. 2. This is made strongly secure by setting $y = g^{rH_2(pw)}$ where $r \in \mathbb{Z}_q^*$ which made public for verification of signature. Conference-key sharing (CONF) [23] assumption states that given $(g, g^a, g^{ab})$, compute $g^b$, where $g \in \mathbb{G}_1$ and $a, b \in_R \mathbb{Z}_q^*$, is hard to achieve [22]. Thus given $g$, $g^r$, $g^{rH_2(pw)}$, getting $g^{H_2(pw)}$ is hard. In ss-PBBS, $g^r$ is not public and only $g$, $g^{rH_2(pw)}$ are public and hence the hardness of solving this is more than CONF. Eventhough $g^{rH_2(pw)}$ is public, offline password guessing attacks will not be effective because it is not possible to distinguish $r$ and $H_2(pw)$ from $rH_2(pw)$. Since $H_2(pw)$ cannot be obtained by enumerating the values of $rH_2(pw)$ and thus finding $pw$ is hard.

**Verification** algorithm(Verify($m, \sigma, y_2, y$)) helps to verify the validity of the message-signature pair.

```
if $e(\sigma, g) \overset{?}{=} e(H_1(m), y_2 y)$
   $return\ 1$
else $return\ 0$
```

To show the **correctness** of verification algorithm(Verify($m, \sigma, y_2, y$)), the equation can be expanded as follows.

Note that $\sigma = \frac{\sigma' L^{rH_2(pw)} H_1(m)^\eta}{(y_1 y_2)^k}$

$$\begin{aligned}
&= \frac{L^{x_2-\eta} L^{rH_2(pw)} H_1(m)^{rH_2(pw)-x_1}}{(g^{x_1}g^{x_2})^k} \\
&= \frac{L^{x_2+rH_2(pw)-\eta} H_1(m)^{rH_2(pw)-x_1}}{(g^{x_1}g^{x_2})^k} \\
&= \frac{L^{x_1+x_2} H_1(m)^{rH_2(pw)-x_1}}{g^{k(x_1+x_2)}} \\
&= \frac{(H_1(m)g^k)^{x_1+x_2} H_1(m)^{rH_2(pw)-x_1}}{g^{k(x_1+x_2)}} \\
&= H_1(m)^{x_1+x_2+rH_2(pw)-x_1} \\
&= H_1(m)^{x_2+rH_2(pw)}
\end{aligned}$$

**Setup**($1^\kappa$): Select a pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ where $\mathbb{G}_1$ and $\mathbb{G}_2$ are cyclic prime order group in $q$ and a generator $g \in \mathbb{G}_1$. Select hash functions, $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$ and $H_2 : \{0,1\}^* \rightarrow {\mathbb{Z}_q}^*$ and return public parameters $params \leftarrow (e, q, \mathbb{G}_1, \mathbb{G}_2, g, H_1, H_2)$.

```
USER                                                     SIGNER

KeyGen_U($pw$):                                          KeyGen_S($\eta$):
  $x_1 \leftarrow_R \mathbb{Z}_q^*$, $r \in_R \mathbb{Z}_q^*$                      $x_2 \leftarrow_R \mathbb{Z}_q^*$
  $y_1 \leftarrow g^{x_1}$                                           $y_2 \leftarrow g^{x_2}$
  $y \leftarrow g^{rH_2(pw)}$
  $\eta \leftarrow rH_2(pw) - x_1$          $\xrightarrow{\eta}$          Signing Key, $sk_{PBBS} = x_2 - \eta$
  $return(x_1, y_1, y, \eta)$                              $return(x_2, sk_{PBBS}, y_2)$

Secret Keys(sk): $sk_{\mathcal{U}} = x_1, sk_{\mathcal{S}} = x_2$,   Public Keys(pk): $pk_{\mathcal{U}} = y_1, pk_{\mathcal{S}} = y_2$, $y = g^{rH_2(pw)}$

Request_U($m, pk, pw$):
  $k \leftarrow_R \mathbb{Z}_q^*$
  $L = H_1(m)g^k$                   $\xrightarrow{L}$           Issue_S($L, pk, sk_{PBBS}$):
  $state \leftarrow (m, k, pw)$                                 $\sigma' = (L)^{sk_{PBBS}}$
                                    $\xleftarrow{\sigma'}$
Unblind_U($\sigma', pk, state$):
  if ($e(L, y_2 g^{-\eta}) \overset{?}{=} e(\sigma', g)$)
    then
      $\sigma = \dfrac{\sigma' L^{rH_2(pw)} H_1(m)^\eta}{(y_1 y_2)^k}$
      if $Verify(m, \sigma, y_2, y) = 1$
        then $return(\sigma)$
  $return$ ($\perp$)
```

Fig. 2. Strongly secure password based blind signature scheme(ss-PBBS)

Therefore,

$$\begin{aligned} e(\sigma, g) &= e(H_1(m)^{x_2 + rH_2(pw)}, g) \\ &= e(H_1(m), g^{x_2} g^{rH_2(pw)}) \\ &= e(H_1(m), y_2 y) \end{aligned}$$

## A. *Proof of Security*

The security of ss-PBBS scheme can be proved in consideration with the properties of unforgeability, blindness and unframeability. The following theorems show that proposed ss-PBBS scheme is perfectly unforgeable, blind and unframeable in the random oracle under computational Diffie Hellman(CDH) assumption.

***Theorem 1:*** **The strongly secure password based blind signature is existentially unforgeable against adaptive chosen message attack(EUF-CMA) under CDH assumption with an advantage of challenger at least $\epsilon/e(1 + q_I)$.**

**Proof**:- In this simulation game adversary($\mathcal{A}$) plays the role of user($\mathcal{U}$) and the challenger($\mathcal{C}$) as that of the signer($\mathcal{S}$). The approach to security proof is similar to [16] and is as follows. If there exists an adversary $\mathcal{A}$ who can break the scheme, then there will be a challenger $\mathcal{C}$ who can make use of $\mathcal{A}$ to solve the CDH which is considered to be a hard problem.

- **Setup Phase**: Challenger chooses public system parameters $(e, q, \mathbb{G}_1, \mathbb{G}_2, g, H_1, H_2)$ in which $H_1$ and $H_2$ are $cryptographic\ hash\ functions$ which behave as random oracle. $\mathcal{C}$ sets $y_2 = g^a$ which is considered to be the public key of the signer($pk_{\mathcal{S}}$) and sends public parameters and $y_2$ to $\mathcal{A}$.
- **Training Phase**: During this phase $\mathcal{A}$ is permitted to access the following oracles.

  – $H_1$**-Oracle:** $H_1$-Oracle works in the following way. An adversary can be able to make $q_{H_1}$ queries with $m_i$ and the challenger should be able to respond back to these queries with $h_i$. $\mathcal{C}$ maintains $H_1$-list and this will be empty initially. When $\mathcal{A}$ queries the oracle with $m_i$, $\mathcal{C}$ responds as follows.
  If the query comes with $m_i$, it checks whether it is in the $H_1$-list. If it is present in the $H_1$-list as a tuple $(hcoin_i, m_i, h_i, u_i)$, then $\mathcal{C}$ replies with $h_i$ from the list. Otherwise, $\mathcal{C}$ flips a coin randomly where $hcoin \in \{0,1\}$, which gives 1 with probability $\alpha$ and 0 with probability $1 - \alpha$. $\mathcal{C}$ also randomly chooses $u_i \in_R \mathbb{Z}_q^*$ and makes the $H_1$-list tuple as follows.
  **1.** If $hcoin = 0$, $\mathcal{C}$ sets $h_i = H_1(m_i) = g^{u_i}$ and insert the tuple $(hcoin_i, m_i, h_i, u_i)$ in to the $H_1$-list. Give $h_i$ to $\mathcal{A}$.
  **2.** Else, sets $h_i = H_1(m_i) = g^{u_i} g^b$ and insert the tuple $(hcoin_i, m_i, h_i, u_i)$ in to the $H_1$-list. Respond this $h_i$ as answer to the query by $\mathcal{A}$.

- $H_2$**-Oracle:** An adversary can be able to make $q_{H_2}$ queries with $pw_j$ and the challenger should be able to respond back to these queries. It is done by maintaining a $H_2$-list which is initially empty. When $\mathcal{A}$ queries the oracle with $pw_j$, $\mathcal{C}$ randomly take $w_j \in_R \mathbb{Z}_q^*$ and give $H_2(pw_j) = w_j$. Challenger also randomly selects $r_j \in_R \mathbb{Z}_q^*$ and stores $(pw_j, w_j, r_j)$ in the $H_2$-list and later if the query appears with $pw_j$ in the $H_2$-list, then gives the same $w_j$ from the tuple $(pw_j, w_j, r_j)$ to the adversary.
- **Issue Oracle:** In the unforgeability game, adversary $\mathcal{A}$ can access the $Issue$ oracle also. The signature is forgeable if the user is able to sign the message without the participation of the signer. Therefore, signer's privacy should be maintain in the unforgeability game rather than the privacy of the user. $\mathcal{A}$ chooses $m_i$ and $pw_j$ and requests the challenger $\mathcal{C}$ for the signature on message $m_i$ with password $pw_j$. $r_j$ and $w_j = H_2(pw_j)$ will be obtain from $H_2$-list, if it is already queried. Otherwise, $\mathcal{C}$ randomly take $w_j \in_R \mathbb{Z}_q^*$, $r_j \in_R \mathbb{Z}_q^*$ and give $H_2(pw_j) = w_j$ and store it as the tuple$(pw_j, w_j, r_j)$ in the $H_2$-list. Then $\mathcal{C}$ checks that whether $m_i$ is queried or not.
  **1.** If $m_i$ is queried, $\mathcal{C}$ retrieve the corresponding tuple $(hcoin_i, m_i,\ h_i, u_i)$ from the $H_1$-list. If $hcoin_i = 0$, $\mathcal{C}$ calculates and outputs $\sigma_i = y_2^{u_i}(h_i)^{r_j w_j}$. If $hcoin_i = 1$ then $\mathcal{C}$ aborts and reports failure.
  **2.** If $m_i$ is not queried, $\mathcal{C}$ runs the $H_1$-$Oracle$ to get the $h_i$, $hcoin_i$ and $u_i$ values and insert these values in $H_1$-list. Then by using these values, produce the signature according to step 1 in $Issue\ Oracle$.

- **Forgery Phase**: On getting sufficient training, $\mathcal{A}$ produces a message-signature pair $(m^*, \sigma^*)$ for a specific $pw_j$ such that $m^*$ is not queried to $Issue\ Oracle$ and $\sigma^*$ is valid. But $m^*$ should be queried to $H_1$-$Oracle$ and $\mathcal{C}$ obtains the tuple $(hcoin^*, m^*,\ h^*, u^*)$ from $H_1$-list. If $m^*$ is not queried to $H_1$-$Oracle$ abort. From $H_2$-$Oracle$ $\mathcal{C}$ obtains $r_j$ since the tuple consists of $(pw_j, w_j, r_j)$. If $m^*$ is queried, then in some cases $\mathcal{C}$ can solve hard problem(here CDH) as follows.
  If $hcoin^* = 0$, $\mathcal{C}$ cannot do much and responds as simulation failure. But, if $hcoin^* = 1$, $\mathcal{C}$ can solve the CDH problem as follows. First $\mathcal{C}$ returns $h^*$ and $u^*$ from $H_1$-list and then compute $g^{ab}$ as follows.
$$\frac{\sigma^*}{y_2^{u^*}\ (h^*)^{r_j w_j}} = \frac{(h^*)^{a + r_j H_2(pw_j)}}{(g^a)^{u^*}\ (h^*)^{r_j w_j}} = \frac{(g^{u^*} g^b)^a\ (h^*)^{r_j H_2(pw_j)}}{(g^a)^{u^*}\ (h^*)^{r_j H_2(pw_j)}} = g^{ab}$$
  This solves CDH problem which is a contradiction to CDH assumption. This indicates that $\mathcal{A}$ cannot produce a valid signature $\sigma^*$ for the message $m^*$. Thus, we can say that there is no forgery possible in polynomial time with non negligible advantage.

**Probability Analysis:** In the proof of Theorem 1, challenger needs to abort the game in certain situations. The requirement is that the probability of aborting is to be negligible. Suppose adversary makes a total of $q_I$ issue queries. As mentioned earlier, let $hcoin \in \{0,1\}$,be 1 with probability $\alpha$ and 0 with probability $1 - \alpha$. During simulation $hcoin = 1$ is the abort condition in training phase and $hcoin = 0$ is the abort condition in challenge phase. Therefore, the probability that challenger does not abort in training phase is $(1-\alpha)^{q_I}$. The probability that challenger does not abort in forgery phase is $\alpha$. Let challenger does not abort during training phase is $E_1$ and challenger does not abort during forgery phase is $E_2$

Pr(Challenger does not abort during simulation)=$Pr(E_1) \wedge Pr(E_2)$
Therefore,
Pr(Challenger does not abort during simulation)=$\alpha(1-\alpha)^{q_I}$. By maximizing this value at $\alpha_{opt} = 1 - 1/(q_I + 1)$, probability that challenger does not abort during simulation is at least $1/e(1+q_I)$ which is non negligible, where $q_I$ is the number of issue queries. Therefore, we can conclude that the advantage of challenger is at least $\epsilon/e(1+q_I)$ as required. This probability analysis technique is similar to [19], where the authors use an approach similar to Coron's analysis [20] of the full domain hash signature scheme.

***Theorem 2:*** **The strongly secure password based blind signature satisfies blindness such that it is infeasible for a malicious signer to distinguish between the two messages $m_0$ and $m_1$ has been signed first in two executions with the honest user.**

**Proof**:- In this game the role of adversary $\mathcal{A}$ and challenger $\mathcal{C}$ is interchanged from the above game. $\mathcal{A}$ provides public parameters($params$) and two messages $m_0, m_1 \in \mathbb{M}$ and sends to $\mathcal{C}$. A random bit $b \in \{0,1\}$ is chosen by the $\mathcal{C}$ and order the messages as $m_b$ and $m_{1-b}$ based on the value of the selected bit '$b$'. The random bit '$b$' is hidden from $\mathcal{A}$. $\mathcal{A}$ has given black box access to two oracles $\mathcal{U}(params, pk, m_b)$ and $\mathcal{U}(params, pk, m_{1-b})$. This $\mathcal{U}$ algorithms perform PBBS protocol and produce the outputs $\sigma_b$ and $\sigma_{1-b}$ corresponds to $m_b$ and $m_{1-b}$. If $\sigma_b \neq \perp$ and $\sigma_{1-b} \neq \perp$ then only $\mathcal{A}$ receives $(\sigma_0, \sigma_1)$. If $\sigma_b = \perp$ and $\sigma_{1-b} \neq \perp$ then $\mathcal{A}$ receives $(\perp, \epsilon)$. If $\sigma_b \neq \perp$ and $\sigma_{1-b} = \perp$ then $\mathcal{A}$ receives $(\epsilon, \perp)$. If $\sigma_b = \perp$ and $\sigma_{1-b} = \perp$ then $\mathcal{A}$ receives $(\perp, \perp)$. After accessing the black boxes $\mathcal{A}$ tries to predict '$b$' and we prove that $\mathcal{A}$ can do this with negligible advantage. That is, there is only guessing probability.
Challenger selects $k$ randomly from $\mathbb{Z}_q^*$ and sends $L$ to $\mathcal{A}$ where $L = H_1(m_b)g^k$ which is uniformly distributed in $\mathbb{G}_1$. $\mathcal{A}$ returns back $\sigma' \in \mathbb{G}_1$ to the first oracle($\mathcal{U}(params, pk, m_b)$) and chooses the value using any strategy he/she wants. At this point $\mathcal{A}$ fixes on the value and he/she is able to predict the output $\sigma_i$ of the oracle $\mathcal{U}(params, pk, m_b)$ with negligible advantage as follows.
Step 1: $\mathcal{A}$ checks if $e(L, y_2 g^{-\eta}) = e(\sigma', g)$ holds. If the check fails, record $\sigma_b$ as $\perp$. Otherwise record the value as $\sigma_b$.
Step 2: Similar to above $\mathcal{A}$ chooses any value $\sigma' \in \mathbb{G}_1$ for the second oracle and do the similar check. If the check fails, record $\sigma_{1-b}$ as $\perp$. Otherwise record the value as $\sigma_{1-b}$.
Step 3: If $\sigma_b = \perp$ and $\sigma_{1-b} \neq \perp$ then output $(\perp, \epsilon)$. If $\sigma_b \neq \perp$ and $\sigma_{1-b} = \perp$ output $(\epsilon, \perp)$. If both checks fails then output $(\perp, \perp)$. If anyone of these three cases occurs, abort.
Step 4: Finally the adversary, $\mathcal{A}$ could predicts $(\sigma_b, \sigma_{1-b})$ only

if $\sigma_b \neq \perp$ and $\sigma_{1-b} \neq \perp$. That is, if both check succeeds then $\mathcal{A}$ initiates PBBS protocol on $m_b$ and $m_{1-b}$ and outputs $\sigma_b, \sigma_{1-b}$ respectively. If either protocol run fails, abort.
This prediction is true because $\mathcal{A}$ performs the same check as that of honest user. If $\mathcal{A}$ is able to predict the final output of its oracles accurately, then $\mathcal{A}$'s advantage in distinguishing $\mathcal{U}(params, pk, m_b)$ and $\mathcal{U}(params, pk, m_{1-b})$ is the same without this final output. Therefore, all of $\mathcal{A}$'s advantage to distinguish between these signatures must come from distinguishing the earlier message of the oracles($L$). These oracles send only uniformly random values and hence $\mathcal{A}$ cannot distinguish between them with non-negligible probability. Therefore we can define adversary $\mathcal{A}$'s advantage in the game as $|Pr[b' = b] - 1/2|$.

***Theorem 3:*** **If CDH assumption holds, the strongly secure password based blind signature provides unframeability under random oracle.**

**Proof**:- To prove the unframeability, signer should not be able to create a signature on behalf of the user without finding user's password. We can prove the security of the scheme under CDH assumption. In this simulation game signer plays as adversary and user as challenger.

- **Setup Phase**: Challenger $\mathcal{C}$ sets $y = g^a$ where $a = rH_2(pw)$. $\mathcal{C}$ sends public parameters and $y$ to $\mathcal{A}$.
- **Training Phase**: During this phase $\mathcal{A}$ has access to $Request$ and $Unblind$ oracles along with $H_1$-$Oracle$.
  - **$H_1$-Oracle:** This hash oracle is similar to that of $H_1$-oracle in the security proof of $Theorem$ 1 with only difference is that it is provided by the user.
  - **Request Oracle:** In this phase $\mathcal{A}$ selects $m_i$ and queries for signature request,$L$ from the $\mathcal{C}$. It can be simulated as follows.
    **1.** If $m_i$ is queried, $\mathcal{C}$ retrieve the tuple $(hcoin_i, m_i, h_i, u_i)$ corresponds to $m_i$ from the $H_1$-list. $\mathcal{C}$ randomly selects $k \in_R \mathbb{Z}_q^*$ and computes $L = h_i g^k$ where $h_i = H_1(m_i)$. $\mathcal{A}$ gets $L$ as output from the $Request\ Oracle$.
    **2.** If $m_i$ is not queried, run the $H_1$-$Oracle$ and gets $h_i$ corresponds to $m_i$ and do the similar step as above.
    Here the $Request\ Oracle$ is similar to the normal $Request$ algorithm. $Unblind\ Oracle$ can be simulated as follows.
  - **Unblind Oracle:** $\mathcal{A}$ queries this oracle with a message,$m_i$.
    **1.** If $m_i$ is queried, $\mathcal{C}$ retrieve the tuple $(hcoin_i, m_i, h_i, u_i)$ corresponds to $m_i$ from the $H_1$-list. Then, if $hcoin_i = 0$, $\mathcal{C}$ calculates and outputs $\sigma_i = (y\ y_2)^{u_i}$. If $hcoin_i = 1$ then $\mathcal{C}$ aborts and reports failure.
    **2.** If $m_i$ is not queried, run the $H_1$-$Oracle$ and insert the tuple $(hcoin_i, m_i, h_i, u_i)$ in to the $H_1$-list. Then produce the signature according to step 1 in $Unblind\ Oracle$.
- **Frameability Phase**: After getting sufficient training, $\mathcal{A}$ produces a message-signature pair $(m^*, \sigma^*)$ such that such that $m^*$ is not queried to $Request$ and $Unblind\ Oracle$ and $\sigma^*$ is valid. But $m^*$ should be queried to $H_1$-$Oracle$ and $\mathcal{C}$ obtains the tuple $(hcoin^*, m^*, h^*, u^*)$ from $H_1$-list. If $m^*$ is not queried to $H_1$-$Oracle$ abort. If $m^*$ is queried, then in some cases challenger $\mathcal{C}$ can solve hard problem(here again CDH) as follows.
  If $hcoin^* = 0$, $\mathcal{C}$ cannot do much and responds as simulation failure. But, if $hcoin^* = 1$, $\mathcal{C}$ can solve the CDH problem as follows. First $\mathcal{C}$ returns $u^*$ from $H_1$-list and then compute $g^{ab}$ and $g^{bx_2}$ as follows.
$$\begin{aligned}\frac{\sigma^*}{(y\ y_2)^{u^*}} &= \frac{(h^*)^{a+x_2}}{y^{u^*}\ g^{x_2 u^*}} \\ &= \frac{(g^{u^*} g^b)^a\ (g^{u^*} g^b)^{x_2}}{(g^a)^{u^*}\ (g^{x_2})^{u^*}} \\ &= g^{ab}\ g^{bx_2}\end{aligned}$$
  $\mathcal{C}$ knows $(g, g^a, g^b, g^{x_2})$ only and compute $g^{ab}$ and $g^{bx_2}$ is known to be CDH problem which is considered to be hard problem. Till today, there is no polynomial time algorithm exists for solving CDH problem. This indicates that $\mathcal{A}$ cannot produce valid signature $\sigma^*$. Thus, we can say that there is no frameability possible in polynomial time with non negligible advantage or the scheme is unframeable.

The probability analysis of Theorem 3 is similar to Theorem 1.

| Scheme | Underlying Signature | Hardness Assumption | Signature Size |
|---|---|---|---|
| *Gjosteen et al. Scheme 1* [9] | RSA | RSA Inversion | 1024 bits |
| *Gjosteen et al. Scheme 2* [9] | CL | LRSW | $2\kappa^*$ bits |
| *PBBS Scheme* [21] | BLS | CDH | 170 bits(constraint in password size) |
| *ss-PBBS Scheme* | BLS | CDH | 170 bits(no constraint in password size) |

$^*\kappa$ is security parameter

TABLE I
COMPARISON WITH EXISTING SCHEMES

## *B. Advantages*

Since the scheme(ss-PBBS) is using both signer's secret key and user's password, it provides more stronger security and it has more efficiency than the existing schemes [9] as shown in Table 1. There is no constraint for the password size and the scheme is not susceptible to offline-password guessing attacks. Thus ss-PBBS scheme is more suitable for client server applications especially for banking applications where both customer and bank secret information are needed for transaction without any password guessing attack.

## VI. CONCLUSION

ss-PBBS scheme is strongly secure scheme and is not susceptible to off-line password guessing attack even if the password size is small. Security proof for this scheme in standard model is an open problem. The scheme can also be made to a honest-user unforgeable password based blind signature scheme using the generic transformation given in [8].

## References

[1] Mihir Bellare and Phillip Rogaway. ”The Exact Security of Digital Signatures - How to Sign with RSA and Rabin”. In *EUROCRYPT*, pages 399-416, 1996.

[2] Claude E. Shannon. ”A mathematical Theory of Communication”. In Bell System Technical Journal, Vol. 27, pp. 379423, October, 1948.

[3] Claude E. Shannon. ”Prediction and Entropy of Printed English”. In Bell System Technical Journal, Vol. 30, n. 1, pp. 50-64, 1951.

[4] William E. Burr, Donna F. Dodson and W. Timothy Polk. ”Information Security”. Electronic Authentication Guideline, Recommendations of the National Institute of Standards and Technology(NIST), Special Publication 800-63, Version 1.0.2, April, 2006.

[5] David Chaum. ”Blind Signatures for Untraceable Payments”. In *Advances in Cryptology - Crypto '82*, Lecture Notes in Computer Science, pages 199-203, Springer, 1983.

[6] Whitfield Diffie and Martin E. Hellman. New Directions in Cryptography. In *IEEE Transactions on Information Theory*, Volume 22(6), pages 644-654, 1976.

[7] Ronald L. Rivest, Adi Shamir and Leonard M. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. In *Communications of the ACM*, Volume 21(2), pages 120-126, 1978.

[8] Dominique Schroder and Dominique Unruh. ”Security of Blind Signatures Revisited”. In *Public Key Cryptography*, Lecture Notes in Computer Science, pages 662-679, Springer, 2012.

[9] Kristian Gjosteen and Oystein Thuen. ”Password-Based Signatures”. In *EuroPKI*, Lecture Notes in Computer Science, pages 17-33, Springer, 2011.

[10] Ari Juels, Michael Luby and Rafail Ostrovsky. ”Security of Blind Digital Signatures (Extended Abstract)”. In *CRYPTO*, Lecture Notes in Computer Science, pages 150-164, Springer, 1997.

[11] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. ”A digital Signature Scheme Secure Against Adaptive Chosen-message attacks”. *SIAM J. Comput.*, 17(2):281-308, April 1988.

[12] David Pointcheval and Jacques Stern. ”Provably Secure Blind Signature Schemes”. In *ASIACRYPT*, pages 252-265, 1996.

[13] David Pointcheval and Jacques Stern. ”Security Proofs for Signature Schemes”. In *EUROCRYPT*, pages 387-398, 1996.

[14] David Pointcheval and Jacques Stern. ”Security Arguments for Digital Signatures and Blind Signatures”. In *J. Cryptology*, 13(3):361-396, 2000.

[15] Jan Camenisch and Anna Lysyanskaya. ”Signature Schemes and Anonymous Credentials from Bilinear Maps”. In *CRYPTO 2004*, pages 56-72, 2004.

[16] Jianhong Zhang and Xiuna Su. ”Another Efficient Blind Signature Scheme based on Bilinear Map”. In *6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*, pages 1-4, 2010.

[17] Dan Boneh, Ben Lynn and Hovav Shacham. ”Short Signatures from the Weil Pairing”. In *ASIACRYPT*, Lecture Notes in Computer Science, pages 514-532, Springer, 2001.

[18] Alexandra Boldyreva. ”Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme”. In *Public Key Cryptography*, Lecture Notes in Computer Science, pages 31-46, Springer, 2003.

[19] Dan Boneh and Matthew Franklin. Identity-Based Encryption from the Weil Pairing. *SIAM Journal of Computing*, Volume 32(3), pages 586615, 2003.

[20] Jean-Sébastien Coron. ”On the Exact Security of Full Domain Hash”. In *CRYPTO 2000*, Lecture Notes in Computer Science, pages 229-235, Springer, 2000.

[21] Sangeetha Jose, Preetha Mathew K. and C. Pandu Rangan. ”Password Based Blind Short Signature for Real Time Applications”. In *Central European Conference on Cryptography, CECC'13*, Telc, Czech Republic, June 2013.

[22] Kouichi Sakurai and Hiroki Shizuya. ”Relationships Among the Computational Powers of Breaking Discrete Log Cryptosystems”. In *EUROCRYPT*, Lecture Notes in Computer Science, pages 341-355, Springer, 1995.

[23] Tatsuaki Okamoto. ”Encryption and Authentication Schemes Based on Public-key Systems”. In *Ph.D. Thesis*, The University of Tokyo, 1988.

**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.

**Preetha Mathew K.** is a Ph D scholar in Indian Institute of Technology (IIT) Madras, Chennai, India. She is working under the guidance of Prof C. Pandu Rangan. Her areas of interest focus on provably secure post quantum cryptosystems, especially in code-based cryptosystem and the security issues in cloud computing. Preetha Mathew K. can be contacted at preetha.mathew.k@gmail.com.

**C. Pandu Rangan** is a 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, security issues in cloud computing, algorithms and data structures, game theory, graph theory and distributed computing. C. Pandu Rangan can be contacted at prangan55@gmail.com.