Chapter 7

Pairing-free Identity-based Partially Blind Signature Scheme

Blind signature scheme enables the user to obtain a signature from the signer however, the signer does not know the message he signs and the final signature generated by the user. A partially blind signature (PBS) scheme allows the signer to explicitly include common information in the blind signature under some agreement with the user but, without violating the blindness property. Recently, many PBS schemes have been proposed either by using PKI or bilinear pairing with MTP hash function. The PKI-based PBS scheme has public key certificate management problem, and bilinear pairing and MTP function are time-consuming operations. Thus, identity-based PBS (ID-PBS) scheme without pairing is more efficient for real-life applications as it requires low computation cost and supports easy implementation. This chapter proposed an ID-PBS, which has been analyzed using random oracle model and it is found that the scheme is provably secure under the ECDLP assumption. Also an application, called online anonymous e-cash system, based on our ID-PBS scheme is developed, in which a bank agrees on a common piece of information with a customer and blindly signs on some messages. It may be noted that our e-cash system achieves unforgeability, unlinkability and non-deniability, and can prevent the double spending of e-cash.

7.1 Introduction

In the open literature, several blind signature and PBS schemes have been proposed by researchers as discussed in Section (2.2.2), which shows that most of the existing PBS schemes are not suitable for practical application as they are implemented either by using PKI or bilinear pairings with a MTP hash function. Thus, the computation time of the pairing-based schemes is expensive. Besides, most of the schemes do not meet some security criteria such as double-spending [119, 121], unforgebility [119, 120], partial blindness [117, 118], non-repudiation [122, 123] occur in e-cash system. Therefore, an ID-PBS without these two operations but, having above security feathers must be efficient for e-cash system that uses low-power, low-computing and low-storage devices like smartcard, PDA, cell phone, etc. In 1996, Abe and Fujisaki [29] introduces the notion of PBS scheme that allows a signer to produce a blind signature on a message for a user, which explicitly includes some commonly agreed information that remains clearly visible without violating the blindness property. The PBS on the other hand, overcomes the disadvantage of blind signature scheme such as signer has no control over the attributes except for those related to the public key. Furthermore, the bank does not require distinct public keys for different face values, and the size of the bank's database used to store the e-cashes spent previously would not increase infinitely over time. The bank also includes some common information such as face value, date information, expiration time, etc. into each e-cash.

In general, e-cash system consists of three entities such as bank, customer and merchant, and it requires a protocol for each of the following operations like opening an account, bank registration, e-cash withdrawal, e-cash payment and e-cash deposit. In e-cash system, to acquire some goods or services from a merchant, the user first withdraws an e-cash from a bank, which is then handed over to the merchant. Finally, the merchant verifies the e-cash and deposits it to the bank for further verification. After checking the correctness and the double spending of the e-cash, the bank credited

the merchant's account. One of the most important properties of any e-cash system is the detection and prevention of double-spending of e-cash. Since e-cash is a digital data, which can be copied easily and thus it may be spent more than one time either by dishonest customer or merchant if an efficient scheme is not followed. In an online e-cash scheme, the payment and deposit take place in a single transaction which means that the e-cash is verified by the bank during payment and thus the bank stays online in each transaction, and to prevent the double-spending of e-cash, the merchant must confer with the bank before accepting any e-cash. In case of off-line system, bank stays disconnected and the merchant accepts an e-cash anonymously from the customer, and later on the bank checks the validity of the e-cash submitted. Subsequently, the bank applies some specified efficient mechanisms to identify the double-spending of e-cash deposited earlier. In addition, the bank verifies the e-cash after each transaction, so the risk of double-spending of e-cash is very high in off-line system. Thus, the off-line e-cash is suitable for payments involving small amounts, whereas the online e-cash is applicable requiring large payment amounts.

The remainder of this chapter is organized as follows. Formal definitions and the security properties of an ID-PBS scheme are provided in Sections 7.2 and 7.3, respectively. Section 7.4 describes the proposed ID-PBS scheme and its analysis is given in Section 7.5. Based on the proposed scheme, an efficient online e-cash system and its security discussion are given in Section 7.6. Finally, the concluding remarks are drawn in Section 7.7.

7.2 Definition of an ID-PBS Scheme

The formal definition of a PBS scheme was first proposed by Abe and Okamoto [26], where a signer and a user are assumed to agree on a piece of common information Δ (say) that is composed of the information of the user and the signer. In real-life applications, Δ may be decided by both the user and signer, while in other applications

it may just be sent from the user to the signer. The information Δ as a part of the signature σ (say) is calculated based on the information of the user and signer represented by Δ_1 and Δ_2, respectively. For example, let us assume that Δ is used to add a validity date to a signature, and in that case, Δ_1 would hold the information that the user wants to have a signature with any validity date, and on the other hand, Δ_2 would hold the information that the signer does only sign with a validity period for a week (say). Thus, Δ would then hold the corresponding validity date with the duration of one week. It may be noted that in our proposed ID-PBS scheme, the user and signer must prove their witness using zero-knowledge protocol before initiating the signature issuing protocol. The definition of an ID-PBS scheme is given below.

Definition 7.2.1. *An ID-PBS scheme consists of the following four algorithms: Setup, Extract, Issue and Verify whereas the Issue one composed of five algorithms, called Agree, Commitment, Blind, Sign and Unblind.*

- **Setup:** The PKG runs this probabilistic polynomial time (PPT) algorithm, which takes a security parameter $k \in Z^+$ as input and outputs the system's parameter Ω, master private key msk and master public key mpk. The system's parameter Ω is publicly known, while msk is kept secret by PKG.

- **Extract:** This algorithm is assumed to be run by PKG, which takes the system's parameter Ω, an identity ID_i as input and outputs a private/public key pair (d_i, P_i). The private key d_i is sent securely to the corresponding user ID_i.

- **Issue:** Assume that the signer ID_i issues a blind signature for the user without knowing the original message. Now using this algorithm (by both user and signer) that takes a message $m \in \{0,1\}^*$, the system's parameter Ω, and $(ID_i, d_i, P_{pub}, \Delta)$ as input and outputs the signature σ in the following way:

- ◇ **Agree:** The user negotiates with the signer ID_i whose public key is P_i, on the agreed information Δ to be attached to the signed message m.
- ◇ **Commitment:** On input of a random string r, the signer ID_i makes a commitment R and sends it to the user.
- ◇ **Blind:** On input of two random strings a, b and a message m, the user generates a string h that is used to sign the message m blindly by the signer ID_i, and then h is sent to the signer.
- ◇ **Sign:** On input of a string h and signer's private key d_i, this algorithm outputs a partially blind signature σ', and then σ' is sent to the user.
- ◇ **Unblind:** On input of the partially blind signature σ' and blind factors a, b it outputs the unblinded signature σ.

- **Verify:** This is a deterministic algorithm that takes $(\Omega, \sigma, m, \Delta, ID_i, P_i)$ as input and outputs "1" if the message-signature pair (m, Δ, σ) is valid with respect to (Ω, ID_i, P_i), and "0" otherwise.

7.3 Security Properties of an ID-PBS Scheme

The following properties must be satisfied by an ID-PBS scheme.

1. **Completeness:** Anyone can verify the validity of the partially blind signature (m, Δ, σ) by checking whether *Verify*$(ID_i, P_i, \Delta, m, \sigma) = 1$ holds or not. If so, the partially blind signature (m, Δ, σ) is accepted, otherwise the signature is rejected.

   Definition 7.3.1. *If the user and the signer follow the signature issuing protocol, the signature scheme is complete if, for every constant $n>0$, there exists a bound*

k_0 such that signer outputs completed and Δ, and the user outputs (m, Δ, σ) that satisfies Verify$(ID_i, P_i, \Delta, m, \sigma) = 1$ with probability at least $(1 - 1/k^n)$ for $k > k_0$.

2. **Partial blindness:** In any PBS scheme, one of the important properties is *partial blindness*, which is defined in terms of the following *unlinkability* game played between a challenger $\mathscr{C}$ and an adversary $\mathscr{A}$.

   - **Setup:** The challenger $\mathscr{C}$ takes a security parameter $k \in Z^+$ and runs this *Setup* algorithm to generate the system's parameter Ω. Then, $\mathscr{C}$ sends Ω to the adversary $\mathscr{A}$ and kept msk away from $\mathscr{A}$.
   - **Preparation:** $\mathscr{A}$ chooses two distinct messages m_0 and m_1, together with the agreed information Δ and an identity ID_i, and sends them to $\mathscr{C}$.
   - **Challenge:** $\mathscr{C}$ chooses a random bit $b \in \{0, 1\}$ secretly, and then ask $\mathscr{A}$ to partially sign on the message m_b with agreed information Δ and m_{1-b} with the same information Δ. Now, $\mathscr{C}$ unblinds both signatures, it presents the signature of m_b to the adversary $\mathscr{A}$.
   - **Response:** $\mathscr{A}$ returns the guess bit b' and wins the game if $b' = b$.

   Definition 7.3.2. *An ID-PBS scheme satisfies the partial blindness, if for every constant $n > 0$, there exists a bound k_0, $\mathscr{A}$ outputs the guess bit b' such that $b' = b$ with probability at most $(1/2 + 1/k^n)$ for $k > k_0$.*

3. **Unforgeability:** In any PBS scheme, *unforgeability* is an important property which ensures that other than the signer anyone cannot produce the valid signature on a message. To define the *unforgeability*, let us introduce the following *challenge-response* game among the adversary $\mathscr{A}$ that plays the role of user and the challenger $\mathscr{C}$ who plays the role of honest signer.

- **Setup:** $\mathscr{C}$ takes the security parameter $k \in Z^+$ and runs the *Setup* algorithm to generate the system's parameter Ω, master secret key msk. Then the challenger $\mathscr{C}$ sends the system's parameter Ω to the adversary $\mathscr{A}$.
- **Hash queries:** $\mathscr{A}$ can ask the output of hash function for the selected input.
- **Extract queries:** $\mathscr{A}$ chooses an identity ID_i and sends it to $\mathscr{C}$. $\mathscr{C}$ then computes the private key d_i by executing *Extract* algorithm and sends d_i to $\mathscr{A}$.
- **Issues queries:** $\mathscr{A}$ chooses a message m, an information Δ, an identity ID_i and a corresponding public key P_i. It sends (ID_i, P_i, m, Δ) to $\mathscr{C}$. $\mathscr{C}$ issues the signature σ and sends it to $\mathscr{A}$.
- **Forgery:** Finally, $\mathscr{A}$ outputs a tuple $(m^*, \Delta^*, \sigma^*, ID_i^*, P_i^*)$. This tuple must satisfy the following requirements:
  - ◇ The signature σ^* must satisfies the *Verify* algorithm.
  - ◇ $\mathscr{A}$ has never asked for private key of the signer whose identity is ID_i^*.
  - ◇ The tuple $(m^*, \Delta^*, ID_i^*, P_i^*)$ has never been submitted for *Issue* queries.

Definition 7.3.3. *An ID-PBS scheme is existential unforgeable against the adaptive chosen message and identity attacks if there is no probabilistic polynomial-time bounded adversary who can win the above challenge-response game with a non-negligible advantage.*

7.4 Proposed Pairing-free ID-PBS Scheme

We motivated from Schnorr's signature scheme [25] and identity-based pairing-free schemes proposed in [14, 15, 22, 205, 211, 213, 214], and proposed an efficient and secure pairing-free ID-PBS scheme using ECC. In our settings, there are three entities namely

a trusted authority PKG, a signer B and a user C. For any partially blind signature, B and C are assumed to agree on a piece of common information Δ. The PKG is responsible to generate the system's parameter Ω and helps the signer B to produce a blind signature σ for the user C on some message m and a common information Δ. The proposed ID-PBS scheme consists of the following algorithms: *Setup*, *Extract*, *Issue* and *verify*, as described below, where the following notations have been used as given in Table 7.1.

Table 7.1: Various notations and their meaning used in the proposed scheme.

Notation	Meaning
$C/B/M$	The User (Customer)/Signer (Bank)/Merchant
ID_i	Identity of the entity i
F_q	A prime field of order q
E/F_q	Set of elliptic curve points
G_q	Additive cyclic group of elliptic curve points
q	k-bit prime number and order of the group G_q
P	The generator of the group G_q
x	Private key of PKG
P_{pub}	Public key of PKG, where $P_{pub} = xP$
(d_i, P_i)	Private/public key of the entity ID_i, where $P_i = d_iP$
H_5, H_6	Secure and one-way cryptographic hash functions (e.g., SHA-1)

- **Setup:** This algorithm takes a security parameter $k \in Z^+$ and then returns system's parameter with a master key. Given k, KGC does the following:

  (a) Choose a k-bit prime q and determine the tuple $\{F_q, E/F_q, G_q, P\}$.

  (b) Choose the master key $x \in_R Z_q^*$ and compute the system public key $P_{pub} = xP$.

(c) Choose two cryptographic secure hash functions H_5:$\{0,1\}^* \times G_q \rightarrow Z_q^*$ and H_6:$\{0,1\}^* \times \{0,1\}^* \times G_q \rightarrow Z_q^*$.

(d) Publish $\Omega = \{F_q, E/F_q, G_q, P, P_{pub}, H_5, H_6\}$ as system's parameter and keep the master key x secret.

- **Extract:** It takes system's parameter Ω, the master key x, and B's identity ID_B as input, returns the identity-based private key of B. With this algorithm, PKG works as follows for the signer B with identity ID_B.

  (a) Choose a number $r_B \in_R Z_q^*$, compute $R_B = r_B P$ and $h_B = H_5(ID_B, R_B)$.

  (b) Compute $d_B = r_B + h_B x \bmod q$.

  Now, PKG sends (d_B, R_B) to B using a secure/out-of-band channel. The public key of B is $P_B = R_B + h_B P_{pub}$ and then he can verify his private/public key pair (d_B, P_B) by checking whether the equation $P_B = d_B P = R_B + h_B P_{pub}$ holds. The private key/public key pair is valid if the above equation holds and vice versa (Eq. 5.1).

- **Issue:** Suppose the user C wants to get a signature on a message m from the signer B. The signature issuing protocol is described as follows.

  (a) Agree: Let us assume that the user C and the signer B has already negotiated the common information Δ.

  (b) Commitment: B chooses a number $r \in_R Z_q^*$, computes $R = rP_B$ and sends (R, R_B) to the user C.

  (c) Blind: C chooses two blind factors $a, b \in_R Z_q^*$ and computes $R' = aR + abP + ab[R_B + H_5(ID_B, R_B)P_{pub}] = aR + abP_B + abP$ and $h = a^{-1}H_6(m, R', \Delta) + b$. Then C sends h to the signer B.

  (d) Sign: B computes $S = (r + h)d_B$ and sends it to the user C.

  (e) Unblind: C computes $S' = a(S + b)$ and outputs the resulting partially blind signature (m, Δ, R_B, R', S').

Then (m, Δ, R_B, R', S') is the partially blind signature on the message m.

- **Verify:** To verify the signature (m, Δ, R_B, R', S') for the message m and the common information Δ of a signer B whose identity ID_B and public key P_B, the verifier performs the following:

  (a) Compute $H_6(m, R', \Delta)$.

  (b) Accept the signature (m, Δ, R_B, R', S') if and only if $S'P = R' + H_6(m, R', \Delta)[R_B + H_5(ID_B, R_B)P_{pub}]$, i.e., $S'P = R' + H_6(m, R', \Delta)P_B$ holds.

For further understanding, the proposed scheme is depicted in Fig. 7.1.

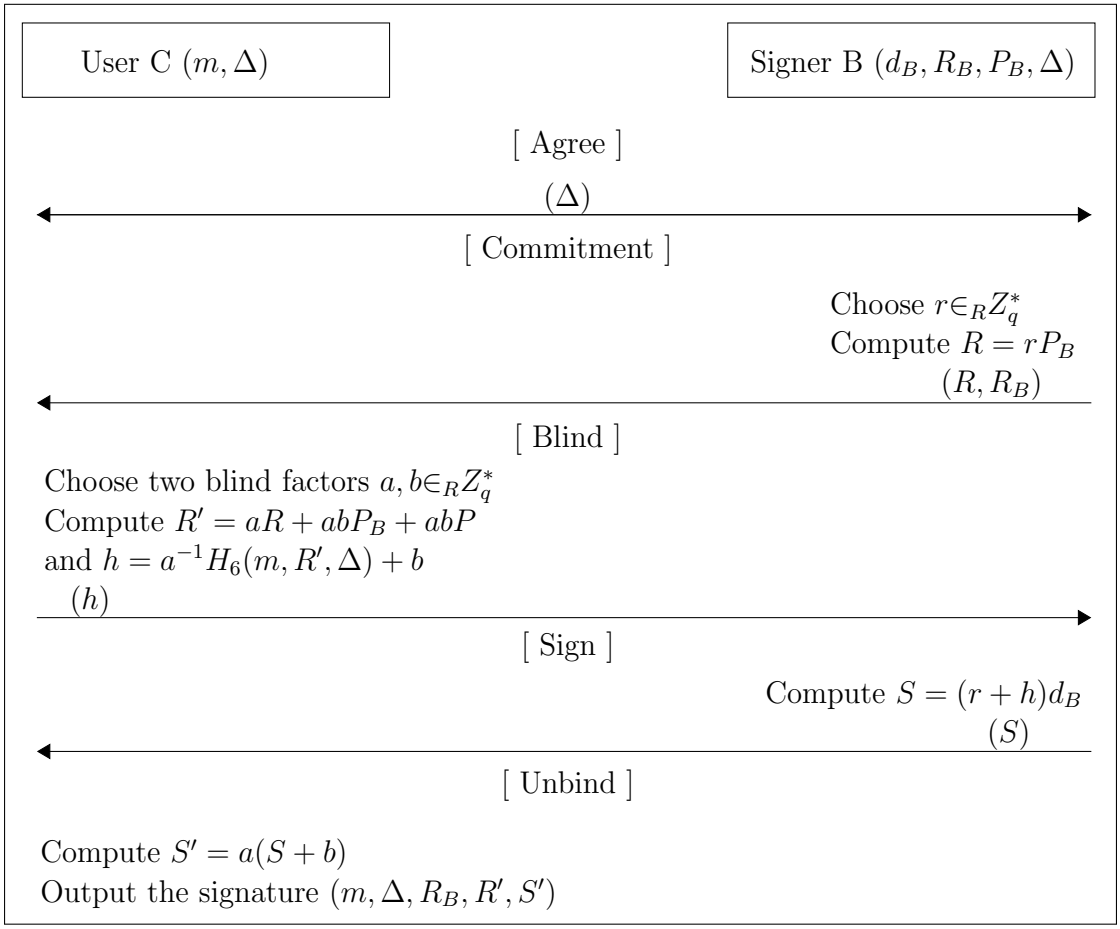

Figure 7.1: Proposed pairing-free ID-PBS scheme.

7.5 Analysis of the Proposed ID-PBS Scheme

7.5.1 Security Analysis

In this section, we analyzed the security of the proposed pairing-free ID-PBS scheme. We show that our proposed scheme satisfies all the security requirements such as *completeness*, *non-deniability*, *partially blindness* and *unforgeability* of partially blind signature scheme. At first we prove completeness property.

Theorem 7.5.1. *The proposed identity-based partially blind signature scheme satisfies the property of completeness.*

Proof. The verification of the received signature is justified by the following equation:

$$S'P = a(S+b)P$$

$$= aSP + abP$$

$$= a(r+h)d_BP + abP$$

$$= a(r + a^{-1}H_6(m, R', \Delta) + b)[r_B + h_Bx] + abP$$

$$= ar[r_BP + h_BxP] + H_6(m, R', \Delta)[r_BP + h_BxP] + ab[r_BP + h_BxP] + abP$$

$$= arP_B + abP_B + H_6(m, R', \Delta)[r_BP + h_BxP] + abP$$

$$= aR + abP_B + abP + H_6(m, R', \Delta)[R_B + h_BP_{pub}]$$

$$= R' + H_6(m, R', \Delta)[R_B + H_5(ID_B, R_B)P_{pub}]$$

$= R' + H_6(m, R', \Delta)P_B$

The correctness of the equation above explains the completeness of the proposed pairing-free ID-PBS scheme. □

Theorem 7.5.2. *The proposed scheme achieves the property of non-deniability.*

Proof. The proposed ID-PBS scheme is non-deniable that is, once the signer B generates the signature (m, Δ, R_B, R', S') for the user C but, later on he cannot deny the signature generation against C. Since the final partially blind signature (m, Δ, R_B, R', S') is formed by using the private key d_B of B and a share information Δ, where $S' = a(S + b) = a(r + h)d_B + ab$. The above equation shows that anyone who does not have the knowledge about B's private key d_B, cannot generate the signature (m, Δ, R_B, R', S'). In addition, the verification equation $S'P = R' + H_6(m, R', \Delta)[R_B + H_5(ID_B, R_B)P_{pub}] = R' + H_6(m, R', \Delta)P_B$ ensures that B's public key P_B must involve in the final verification equation. Thus, B couldn't deny the signature generation on the message m. □

Theorem 7.5.3. *The proposed scheme satisfies the partially blindness property.*

Proof. Let us assume that $\mathscr{A}$ be a probabilistic polynomial-time bounded adversary who plays the role of the signer B and (m, Δ, R_B, R', S') be one of two signatures

subsequently given to $\mathscr{A}$. Let (R, h, S) be data appearing in view of $\mathscr{A}$ during one of the executions of the *Issue* protocol. Therefore, to prove the partially blindness property we show that a given valid signature (m, Δ, R_B, R', S') and any view (R, h, S) of it, there always exists a unique pair of blind factors $\alpha, \beta \in_R Z_q^*$. Assume that $\mathscr{A}$'s identity is ID_B, the corresponding public key is P_B and $\mathscr{A}$ acquires the signatures $\{m_b, \Delta, R_B, \sigma_b = (R'_b, S'_b)\}$ and $\{m_{1-b}, \Delta, R_B, \sigma_{1-b} = (R'_{1-b}, S'_{1-b})\}$ during the execution of the *Issue* protocol. Let (R_i, h_i, S_i) for $i = 0, 1$ be data appearing in view of $\mathscr{A}$ during one of the executions of the *Issue* protocol. To show the uniqueness of the blind factors $\alpha, \beta \in_R Z_q^*$ that maps (R_i, h_i, S_i) to (R'_j, S'_j) for i, j, $b \in \{0, 1\}$, consider the following equations:

$$R'_j = \alpha R_i + \alpha\beta P + \alpha\beta P_B \tag{7.1}$$

$$h_i = \alpha^{-1} H_6(m, \Delta, R'_j) + \beta \tag{7.2}$$

$$S_i = (r + h_i) d_B \tag{7.3}$$

$$S'_j = \alpha(S_i + \beta) \tag{7.4}$$

$$S'_j P = R'_j + H_6(m, \Delta, R'_j) P_B \tag{7.5}$$

From the above, α and β can be computed uniquely from equations (7.4) and (7.2) as $\alpha = S'_j/(S_i + \beta)$ and $\beta = (h_i S'_j - S_i H_6(m, \Delta, R'_j))/(H_6(m, \Delta, R'_j) + S'_j)$, respectively. Now, we prove that $\alpha, \beta \in_R Z_q^*$ got from (7.2) and (7.4) satisfy the equation (7.5) with

the help of the equation (7.3). Let $a = S'_j$, $c = h_i$, $b = S_i = (r + c)d_B$ and $d = H_6(m, \Delta, R'_j)$. Therefore, $\beta = \frac{ac-bd}{a+d}$ and $\alpha = \frac{a+d}{b+c}$, and we have

$$
\begin{aligned}
& R'_j + H_6(m, \Delta, R'_j)P_B \\
&= \alpha R_i + \alpha\beta P + \alpha\beta P_B + H_6(m, \Delta, R'_j)P_B \\
&= \frac{a+d}{b+c} r d_B P + \frac{a+d}{b+c}\frac{ac-bd}{a+d} P + \frac{a+d}{b+c}\frac{ac-bd}{a+d} d_B P + d d_B P \\
&= \frac{a+d}{b+c} r d_B P + \frac{ac-bd}{b+c} P + \frac{ac-bd}{b+c} d_B P + d d_B P \\
&= [\frac{a+d}{b+c} r d_B + \frac{ac-bd}{b+c} + \frac{ac-bd}{b+c} d_B + d d_B]P \\
&= [\frac{ard_B + drd_B + ac - bd + acd_B - bdd_B + bdd_B + dcd_B}{b+c}]P \\
&= [\frac{ard_B + drd_B + ac - bd + acd_B + dcd_B}{b+c}]P \\
&= [\frac{ard_B + drd_B + ac - d_B(r+c)d + acd_B + dcd_B}{b+c}]P \\
&= [\frac{ard_B + drd_B + ac - drd_B r + cdd_B + acd_B + dcd_B}{d_B(r+c) + c}]P \\
&= [\frac{ard_B + ac + acd_B}{d_B(r+c) + c}]P \\
&= [\frac{a(rd_B + c + cd_B)}{d_B(r+c) + c}]P \\
&= [\frac{a(rd_B + c + cd_B)}{rd_B + c + cd_B}]P \\
&= aP \\
&= S'_j P \quad [\because \quad a = S'_j]
\end{aligned}
$$

where $m = m_0$ or $m = m_1$. Thus, the blinding factors $\alpha, \beta \in_R Z_q^*$ always exists uniquely, which leads to the same relation as defined in the *Issue* protocol of the proposed scheme. Therefore, even an infinitely powerful adversary $\mathscr{A}$ outputs a guess bit b' such that $b' = b$ with probability $\frac{1}{2}$. Thus, the proposed ID-PBS scheme satisfies the *partially blindness* property. □

Theorem 7.5.4. *The proposed scheme is existential unforgeable in the random oracle model against the adaptively chosen message and identity attacks under the assumption that ECDLP is hard to solve in the elliptic curve group G_q.*

Proof. Assume that the proposed ID-PBS scheme can be forged under the adaptive chosen message and identity attacks by a polynomial-time bounded adversary $\mathscr{A}$, then it is possible to construct an algorithm $\mathscr{C}$, which helps $\mathscr{A}$ to solve an instance of ECDLP that is, $\mathscr{A}$ outputs a from the input tuple $(P, Q = aP)$, where $a \in_R Z_q^*$.

- **Setup:** $\mathscr{C}$ sets the private/public key of PKG as $(x = a, P_{pub} = aP)$. Here, P is the generator of the group G_q and the hash functions H_i (for $i = 5, 6$) are considered as random oracle. $\mathscr{C}$ sets $\Omega = \{F_q,\ E/F_q,\ G_q,\ P,\ P_{pub} = aP,\ H_5,\ H_6\}$ as system's parameter and answer $\mathscr{A}$'s queries in the following way.

- **Extract queries:** $\mathscr{C}$ maintains a H_5-oracle list L_{H5}^{list} that contains the tuples of the

form (ID_i, d_i, R_i, h_i). To obtain the private key of the user whose identity is ID_i, $\mathscr{A}$ makes queries to $\mathscr{C}$, in the following, $\mathscr{C}$ looks for ID_i in the list L_{H5}^{list}, which is initially empty and returns the output to $\mathscr{A}$ as follows:

- ◇ If $(ID_i = ID_B)$, $\mathscr{C}$ outputs "failure" and stops the protocol execution.
- ◇ If $(ID_i \neq ID_B)$, $\mathscr{C}$ selects $a_i, b_i \in_R Z_q^*$ and sets $d_i = b_i, R_i = a_i P_{pub} + b_i P$, and $h_i = H_5(ID_i, R_i) = -a_i$. It is clear that (d_i, R_i) satisfies the equation $P_i = d_i P = R_i + h_i P_{pub}$. Then $\mathscr{C}$ outputs (d_i, R_i) as the secret key of the user ID_i and inserts the tuple (ID_i, d_i, R_i, h_i) to the list L_{H5}^{list} and returns d_i to $\mathscr{A}$.

- **Hash queries to H_6:** $\mathscr{C}$ also maintains an initially-empty H_6-oracle list L_{H6}^{list} that contains the tuple of the form $(m_i, \Delta_i, R_i', h_i)$. Suppose that $\mathscr{A}$ makes at most q_{H6} times H_6 queries. For each query of the form $H_6(m_i, \Delta_i, R_i')$, $\mathscr{C}$ checks the list L_{H6}^{list}:
  - ◇ If there is a tuple $(m_i, \Delta_i, R_i', h_i)$ on L_{H6}^{list}, the algorithm $\mathscr{C}$ sets $H_6(m_i, \Delta_i, R_i') \leftarrow h_i$ and returns it to the adversary $\mathscr{A}$.
  - ◇ If not, that is $H_6(m_i, \Delta_i, R_i')$ has not been queried to H_6-oracle. $\mathscr{C}$ selects a number $h_i \in_R Z_q^*$ such that there is no item $(\cdot, \cdot, \cdot, h_i)$ in L_{H6}^{list}; $\mathscr{C}$ then includes $(m_i, \Delta_i, R_i', h_i)$ to the list L_{H6}^{list} and returns h_i to $\mathscr{A}$.

- **Issue queries:** In this case, $\mathscr{A}$ can make at most q_I times *Issue* queries. For each query of the form $(ID_i, P_i, m_i, \Delta_i)$, $\mathscr{C}$ does the following:

  - ◇ Choose two numbers $S_i', h_i \in_R Z_q^*$.
  - ◇ Set $H_6(m_i, \Delta_i, R_i') \leftarrow h_i$ and store $(m_i, \Delta_i, R_i', h_i)$ to the list L_{H6}^{list}.
  - ◇ Compute $R_i' = S_i'P - h_iP_B$.
  - ◇ Output the signature $(m_i, \Delta_i, R_i, R_i', S_i')$.

- **Forgery:** Finally, $\mathscr{A}$ outputs a valid signature (m, Δ, R_B, R', S'). It follows from the *forking lemma* [212] that if $\epsilon \geq 10(q_I+1)(q_I+q_{H6})/2^k$, then $\mathscr{C}$ which can produce two valid signatures (m, Δ, R_B, R', S') and $(m, \Delta, R_B, R'^*, S'^*)$ on the same message m such that $R' = R'^*$ but $h \neq h^*$.

  Then we can write,

$$\begin{aligned} S'P &= R' + h[R_B + H_5(ID_B, R_B)P_{pub}] \\ &= R' + hP_B \end{aligned} \tag{7.6}$$

$$\begin{aligned} S'^*P &= R' + h^*[R_B + H_5(ID_B, R_B)P_{pub}] \\ &= R' + h^*P_B \end{aligned} \tag{7.7}$$

Subtracting Eq. (7.6) from the Eq. (7.7) and we have

$$\begin{aligned} S'^{*}P - S'P &= R' + h^{*}P_B - R' - hP_B \\ &= h^{*}P_B - hP_B \\ &= (h^{*} - h)P_B \end{aligned} \tag{7.8}$$

Let $R_B = a_B P_{pub} + b_B P = a_B aP + b_B P$. Therefore, from the Eq. (7.8), we get

$$\begin{aligned} (S'^{*} - S')P &= (h^{*} - h)[R_B + h_B aP] \\ &= (h^{*} - h)[a_B aP + b_B P + h_B aP] \\ &= (h^{*} - h)(a_B + h_B)aP + (h^{*} - h)b_B P \end{aligned} \tag{7.9}$$

Now, from the Eq. (7.9), we can solve

$$a = [(S'^{*} - S') - (h^{*} - h)b_B]/[(h^{*} - h)(a_B + h_B)] \tag{7.10}$$

Therefore, $\mathscr{A}$ solves $a = [(S'^{*} - S') - (h^{*} - h)b_B]/[(h^{*} - h)(a_B + h_B)]$. According to the *forking lemma*, $\mathscr{A}$ can solve the ECDLP within the expected time $t' \leq 120686 q_{H6} t/\epsilon$. But the ECDLP is computationally infeasible to the elliptic curve group G_q by any polynomial time-bounded algorithm. Therefore, the proposed ID-PBS scheme is secure under the adaptive chosen message and identity attacks in the random oracle model with the hardness assumption of ECDLP. □

7.5.2 Efficiency Analysis

In this section, we will analyze the proposed ID-PBS scheme in terms of computation costs and compare it with other related schemes. Since the proposed scheme captures the advantages of both IBC and ECC, so the cost of public key certificate management is removed. In addition, our scheme is free from two time-consuming cryptographic operations, namely bilinear pairing and MTP hash function. For computation cost efficiency analysis of our scheme, we used the time complexity notations defined in Table 4.2 (Chapter 4). Note that the proposed scheme executes four T_{ML}, one T_{EA} and one T_{IN} for *Issue* protocol and two T_{ML} and one T_{EA} for signature verification. Thus, the proposed scheme needs $(6T_{ML}+2T_{EA}+T_{IN}) \approx 185T_{ML}$ time totally. Now we compare the proposed scheme with other relevant schemes [26, 116, 128, 129, 210] and summarizes the results in Table 7.2, which shows that the proposed scheme is computationally more efficient than other schemes.

7.6 Application of the Proposed ID-PBS Scheme

The partial blind signatures play the central role in cryptographic protocols to provide the anonymity of users in e-cash system. It allows the signer to explicitly include common information in the blind signature under some agreement with the user but, the signer learns neither the message nor the resulting signature. Several partial blind signatures have been found in the open literature but, all of them can be realized either using PKI or by bilinear pairings with a MTP hash function. Thus, ID-PBS is useful in e-cash system since no public key certificate management is needed but, the computation cost is still high due to the involvement of bilinear pairings and MTP hash function. To achieve the desired level of security and computation efficiency, we proposed an online e-cash scheme using the proposed ID-PBS scheme that is free from bilinear pairings and MTP hash function, in this section.

Table 7.2: Comparison of computation efficiency

Schemes	Computation Cost		Total Computation Cost
	Issue phase	Verification phase	
Abe & Okamoto [26]	$7T_{EX}$	$4T_{EX}$	$11T_{EX} \approx 2640T_{ML}$
Zhang et al. [116]	$4T_{ML} + 3T_{EA} + T_{MTP} + T_{IN}$	$T_{ML} + T_{EA} + T_{MTP} + 2T_{BP}$	$5T_{ML} + 4T_{EA} + 2T_{MTP} + T_{IN} + 2_{TBP} \approx 389T_{ML}$
Chow et al. [114]	$5T_{ML} + T_{EA} + T_{MTP} + T_{IN}$	$T_{ML} + T_{EA} + T_{MTP} + 2T_{BP}$	$6T_{ML} + 2T_{EA} + 2T_{MTP} + T_{IN} + 2T_{BP} \approx 417T_{ML}$
Hu & Huang [210]	$4T_{ML} + 4T_{EA} + T_{MTP}$	$2T_{ML} + T_{EA} + 2T_{BP}$	$6T_{ML} + 5T_{EA} + T_{MTP} + 2T_{BP} \approx 377T_{ML}$
Chen et al. [128]	$5T_{EX}$	$T_{EX} + T_{BP}$	$6T_{EX} + 2T_{BP} \approx 1614T_{ML}$
Zhang et al. [129]	$6T_{ML} + 5T_{EA} + 2T_{BP} + T_{MTP} + 2T_{PX}$	$3T_{BP} + T_{PX}$	$6T_{ML} + 5T_{EA} + T_{MTP} + 5T_{BP} + 3T_{PX} \approx 769T_{ML}$
Proposed	$4T_{ML} + T_{EA} + T_{IN}$	$2T_{ML} + T_{EA}$	$6T_{ML} + 2T_{EA} + T_{IN} \approx 185T_{ML}$

7.6.1 Descriptions

Nowadays the electronic commerce (e-commerce) becomes popular due to its many applications such as electronic payment, electronic funds transfer, financial electronic data exchange, etc. Here we proposed an anonymous online e-cash system based on the proposed ID-PBS scheme. The proposed online e-cash system consists of four entities: (1) a trusted third party, called PKG, (2) the bank (B), (3) the customer (C) and (4) the Merchant (M) and five phases: (1) *Setup phase* (2) *Bank registration*

phase (3) *Opening an account phase* (4) *Withdrawal phase* and (5) *Payment-deposit phase*. We assume that, before initiating a withdrawal phase of e-cash, C and the bank B prove their witness in an interactive manner by means of *zero-knowledge proof* protocol (Section 3.6). With this protocol, C can prove the fact to B that he knows B's secret d_B without revealing it. The PKG is responsible to generate the system's parameter Ω and helps the bank B to issue an e-cash for the customer C. In an e-cash life cycle, B first registers to PKG for the private key. To obtain an e-cash, C opens an account to B. Then, C performs a payment protocol for purchasing some goods from M by using the e-cash that has not been spent previously. On receiving the e-cash, M sends it to B and the bank transfer the corresponding money to M's account provided that the e-cash is valid and fresh. The data flow occurs in an online e-cash system is outlined in Fig. 7.2.

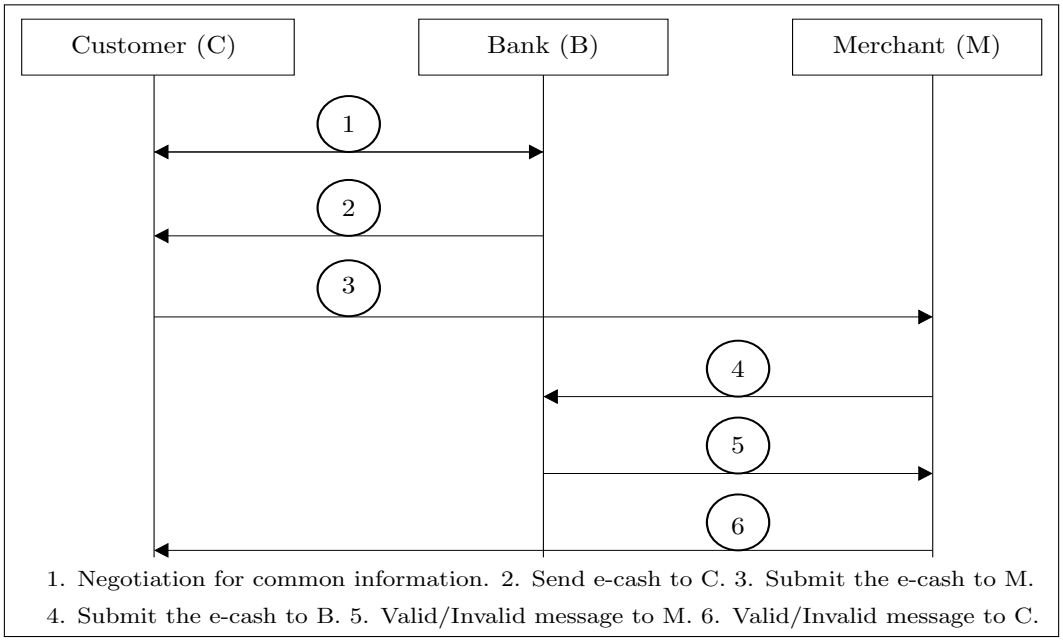

Figure 7.2: Overview of the proposed online e-cash system.

7.6.1.1 Setup Phase

In this phase, the PKG takes a security parameter $k \in Z^+$ as input, returns a master key $x \in_R Z_q^*$, corresponding public key $P_{pub} = xP$ and the system's parameter

$\Omega = \{F_q, E/F_q, G_q, P, P_{pub}, H_5, H_6\}$. Here H_5, H_6 are the general cryptographic hash functions (e.g., SHA-1).

7.6.1.2 Bank Registration Phase

In this phase, the bank B with identity ID_B registers to PKG for the identity-based private/public key pair. The bank B sends his identity ID_B to PKG, then PKG chooses a number $r_B \in_R Z_q^*$, computes $R_B = r_B P$, $h_B = H_5(ID_B, R_B)$ and the private key $d_B = r_B + h_B x$, and then sends (d_B, R_B) securely to B. Therefore, (d_B, P_B) is the private/public key pair of B, where $P_B = R_B + h_B P_{pub}$.

7.6.1.3 Opening an Account Phase

This phase is executed between B and C(customer) when the customer wants to purchase some goods or needs some service from B for which he has to pay some money through the Internet, he will go to the bank to apply for opening an account in advance. For this, C sends the opening account application with some information such as passport number, billing information, etc. from which B can uniquely identify C. The bank B identifies C based on the supplied information and then opens an account ACC_C against the customer C.

7.6.1.4 Withdrawal Phase

In this phase, if C wants to withdraw an e-cash with face value v from B, he then sends his account information ACC_C to B. Then B checks whether ACC_C is valid. If it is, B is ready to withdraw an e-cash for C with the face value v. Before issuing an e-cash, both B and C negotiate a common agreed information Δ. Then C selects a message m and blinds it and then submits the blinded message to B. Then B blindly sign the received blinded message based on explicit common information Δ consisting of the face value v and other information agreed by both parties, such as the expiry date of

the e-cash to be issued. After receiving a blind signature from B, C will unblind it and obtain a regular signature for m, including the face value v. The resulting signature is viewed as e-cash with face value v. The description of withdrawal phase is given below.

- C and B negotiate a common information Δ. The face value v is now attached to the common information Δ.
- B chooses a number $r \in_R Z_q^*$, computes $R = rP_B$ and sends (R, R_B) to C.
- On receiving (R, R_B), C chooses a message m, two blind factors $a, b \in_R Z_q^*$ and computes $h = a^{-1}H_6(m, \Delta, R') + b$, where $P_B = R_B + h_B P_{pub} = d_B P$ and $R' = aR + abP + abP_B$, and C then sends h to B.
- Upon receiving h from C, B computes $S = (r + h)d_B$ and sends it to C.
- Then C computes $S' = a(S + b)$ and outputs the e-cash (m, Δ, R_B, R', S').

Further, we illustrated the e-cash withdrawal phase in Fig. 7.3.

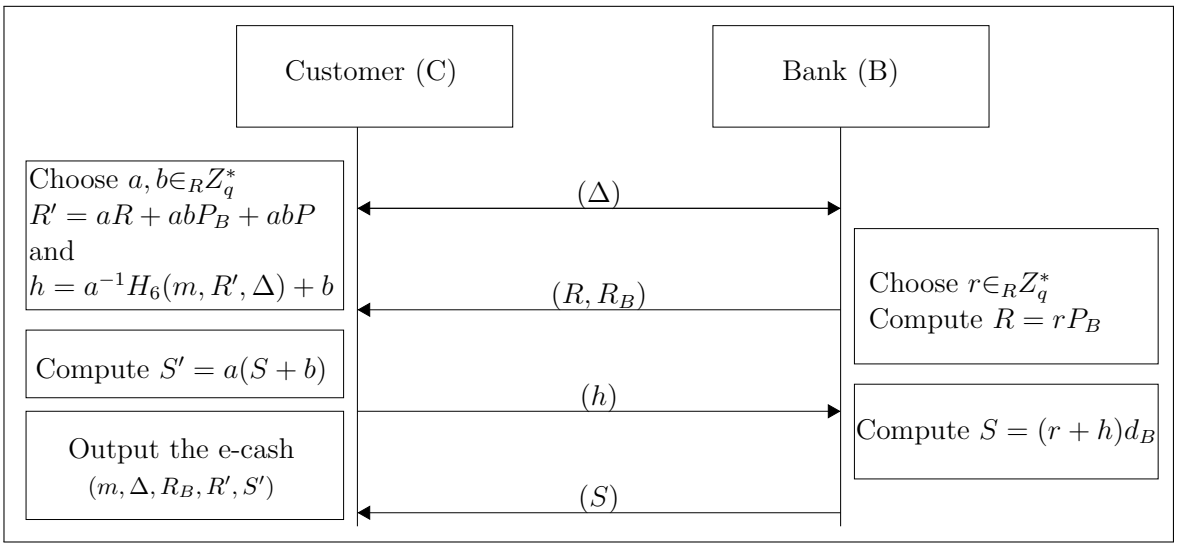

Figure 7.3: Proposed withdrawal phase.

7.6.1.5 Payment-deposit Phase

When C wants to purchase some goods, he sends an e-cash (m, Δ, R_B, R', S') to the merchant M. After receiving the e-cash (m, Δ, R_B, R', S'), M first checks that it is valid by verifying it against the B's public key P_B. If the e-cash (m, Δ, R_B, R', S') is valid, M interacts with B to check the double-spending of it, otherwise sends an error message to C. To prevent the double-spending of (m, Δ, R_B, R', S'), B searches his own database (where the information about the previously spent e-cashes is stored) that m does not exist there. If the e-cash passes both checks, M's account will be credited by amount v. Now, we describe the payment-deposit phase below.

- C sends the e-cash (m, Δ, R_B, R', S') to M.
- Then M checks the validity of received e-cash (m, Δ, R_B, R', S') by verifying whether the equation $S'P = R' + h[R_B + H_5(ID_B, R_B)P_{pub}] = R' + hP_B$ holds.
- If it does not hold, M rejects the process, otherwise, sends the e-cash (m, Δ, R_B, R', S') with his account information to the bank B.
- Then B checks the validity of (m, Δ, R_B, R', S') by executing the above said equation and compare it with the list stored on his database to identify *double-spending* of the received e-cash (m, Δ, R_B, R', S'). If (m, Δ, R_B, R', S') is fresh, B credits M's account by an amount v and sends a validity message to M. Otherwise, B sends a message that indicates that the received e-cash (m, Δ, R_B, R', S') is invalid.
- Depending on the result of above step, M sends Valid/Invalid message to C.

It is to be noted that, in the proposed system, the size of B's database would not increase infinitely over time. Since the face value v, expiry date and time, etc. are included in the common information Δ, so B will not store the information about all e-cashes those have been spent previously. The bank's database contains the information about those e-cashes which are valid with respect to expiry date and time. If B finds

any expired e-cash in its database, he/she simply removes the e-cash from the database. The payment-deposit phase is depicted in Fig. 7.4.

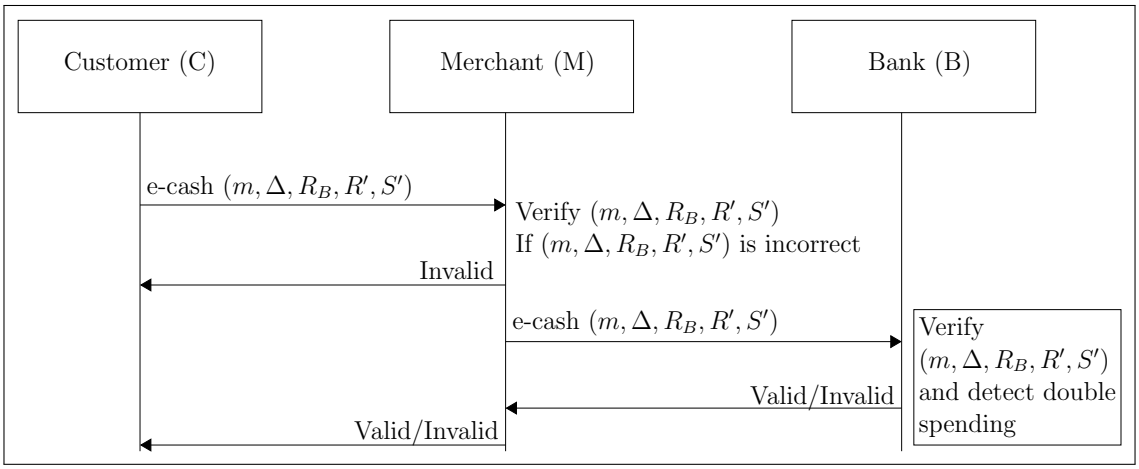

Figure 7.4: Proposed payment-deposit phase.

7.6.2 Analysis of the Proposed Online e-cash System

In the section, we demonstrate the proposed system provides a secure and computation efficient online e-cash system. Our scheme satisfies all the related security of the online e-cash systems.

7.6.3 Anonymity

In a simple e-cash system, *anonymity* is the basic requirement which ensures that, when a customer C draws an e-cash from the bank B and spent it to the merchant M, however, B and M couldn't be able to trace C from the previously spent e-cash. The proposed online e-cash system scheme supports the *anonymity* of customer through the use of partially blind signature. The inherent *unlinkability* of partially blind signature ensures none can determine that the two payments are executed by the same customer. The theorem 7.5.3 shows that the e-cash is *unlinkable*, the bank B wouldn't know anything about C except the face value v of e-cash (m, Δ, R_B, R', S'), since B would

have the record message m of the e-cash (m, Δ, R_B, R', S'). However, B cannot find any link with blinded message he signs and the issued e-cash (m, Δ, R_B, R', S'). Because the blind factors a, b are used to blind the message m and these are unknown to B. Thus, the proposed online e-cash system provides the security of the customer anonymity.

7.6.4 Non-deniability

Non-deniability is also an important property in an e-cash system. This property states that, once B issues an e-cash (m, Δ, R_B, R', S') for C, the bank B later on cannot deny the e-cash generation against the customer C. In the proposed e-cash system, B computes $S = (r + h)d_B$ and C unblinds it as $S' = a(S + b)$. Since, the e-cash (m, Δ, R_B, R', S') is nothing but, a *partially blind signature* that is formed by using the private key d_B of B and a shared information Δ, and B's public key is required in the final verification equation $S'P = R' + h[R_B + H_5(ID_B, R_B)P_{pub}] = R' + hP_B$. Therefore, a valid e-cash (m, Δ, R_B, R', S') can be computed by the bank B only, none can generate it without knowing B's private key d_B. Otherwise, the above verification equation wouldn't be satisfied. Therefore, B cannot repudiate the valid e-cash generation on the message m.

7.6.5 Double-spending Detection

In our proposed system, B can easily detect any doubly spent e-cash, which is a great concern in any e-cash system, using the parameters of that e-cash. On receiving an e-cash (m, Δ, R_B, R', S'), B checks the local database (which contains information about previously spent e-cashes) to see whether m, Δ, R_B, R' and S' are unique. If so, the e-cash (m, Δ, R_B, R', S') is fresh, otherwise it is spent doubly.

7.6.6 Unforgeability

An online e-cash system is unforgeable, if and only if except B, anyone cannot generate the e-cash (m, Δ, R_B, R', S'). The theorem 7.5.4 ensure that e-cash (m, Δ, R_B, R', S') is unforgeable under the adaptively chosen message and identity attacks in the random oracle model with the assumption that ECDLP is intractable.

7.7 Chapter Summary

An ECC-based ID-PBS scheme is proposed in this chapter. Since the scheme has been implemented without bilinear pairings and MTP hash function, it is computation efficient as compared with other related schemes. We prove that our signature scheme is secured against adaptively chosen message and identity attacks in the random oracle model based on ECDLP assumption. Based on our ID-PBS scheme, we design an efficient and secure online e-cash system, which satisfies all the necessary security requirements such as *unforgeability*, *anonymity*, *non-deniability* and detection of *double-spending* of e-cashes. Finally, the proposed online e-cash system has several advantages that suit for real-life applications.

The next chapter i.e., Chapter 8 addresses the design of a password-based remote login scheme for remote user authentication over Internet. The scheme can be used for mutual authentication and session key agreement between a user and the server.