Transferable E-Cash Without Observer

Kamlesh Tiwari

Y7111016

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

July 2009

भारतीय प्रौद्योगिकी संस्थान कानपुर
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

Transferable E-Cash Without Observer

A Thesis Submitted

in Partial Fulfillment of the Requirements

for the Degree of

Master of Technology

by

Kamlesh Tiwari

Y7111016

to the

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

July 2009

भारतीय प्रौद्योगिकी संस्थान कानपुर
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

CERTIFICATE

This is to certify that the work contained in the thesis entitled "*Transferable E-Cash Without Observer*" by "*Kamlesh Tiwari*" has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

July 2009

Prof. Rajat Moona
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur
Kanpur 208016

Prof. Piyush P. Kurur
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur
Kanpur 208016

भारतीय प्रौद्योगिकी संस्थान कानपुर
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

ABSTRACT

The requirement of money to purchase goods and services through Internet and to make electronic payments has become a major thrust area of financial transactions. Use of credit-card or debit-card for payments involves the presence of a third party (bank) at the time of payments, who can use the details of payment for unwanted purposes. E-cash is the right solution for these concerns. It works in offline mode therefore, does not require the presence of any mediator during the payments and reduces the communication overheads over the network.

E-cash promises anonymity, un-linkability, and intractability to its spender until he is genuine and is not involved in any act of misbehavior such as double-spending or over-spending. An acceptable e-cash has to reflect all the properties of real money. There are various models for e-cash offering different properties.

Current transferable e-cash solutions allow a person to spend the cash received from a payer to buy services from another merchant. However before spending the person has to contact an observer to refresh the coin. This solution is very inconvenient. In our work we have proposed a scheme for implementing transferable e-cash that does not require the involvement of any observer during spending. Our solution works upon coin ownership transfer and is built around the restrictive blinding and digital signature scheme. We also proposed modifications in existing e-cash models to reduce their space requirements for suitability to low memory devices like PDA or smart-cards.

भारतीय प्रौद्योगिकी संस्थान कानपुर
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

Acknowledgments

I take this opportunity to express my sincere gratitude to my thesis advisors Prof. Rajat Moona and Prof. Piyush P. Kurur for their excellent guidence and invaluable support. It was an honour working with them. They always boosted my confidence and laid down the foundation on which this work could have been carried out. His vision, commitment and knowledge made this work a truly memorable experience.

I also want to thanks to Mrs. Rajani Moona for sparing her valuable time studying and suggesting improvements to this thesis writeup and providing me morel support.

I also thank to my friends Abhai Khoje, Aditya Nigam, Amit V. Panara, Jignesh, Kamlesh Patel, Mohit K. Dwivedi, M. Rajesh, M. Ravi Babu, Nishant, Prabhat Agnihotri, Rama Kant Pathak, Ravi Dalmiya, Roshan Ali, RAMA Ali, Rahul Kulkarni, Satyam Sharme, Saurabh Joshi, Sameer agrawal, Satendra K. Yadav, Sujith Thomas, Syaam P. Dulla, Vinay Singh, and every one else whose name is missing here only due to lack of space, for making my stay at IIT Kanpur enjoyable.

Finally, I wish to thank my father Shri. K. N. Tiwari, mother Smt. Usha Tiwari, wife Suman, son Utkarsh, brother Gunakesh for their support, encouragement and well-wishes which made this work possible.

भारतीय प्रौद्योगिकी संस्थान कानपुर
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

Dedications

To my Mentors

Dr. S. V. Shukla

Mr. Amit Agrawal

Dr. Anoop Gupta

Mr. Shailendra Agrawal

Mr. L. S. Yadav

Mr. D. D. Singh

Prof. Rajat Moona

Prof. Piyush P. Kurur

गुरुर्ब्रह्मा गुरुर्विष्णुः गुरुर्देवो महेश्वरः।
गुरुः साक्षात्परब्रह्म तस्मै श्री गुरवेनमः॥

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Electronics has made improvements in every aspect of our daily lives. One such revolutionary contribution is the development of paper-less operations and transactions. Due to the rapid growth of digital communication and electronic data exchange, our way of doing commerce has also changed a lot. Electronic payment systems have turned out to be a smart and convenient way of making payments. We often use online digital systems such as the Internet to buy or sell things and card-based systems such as credit cards or debit cards for payments. These activities require a new kind of realization of money suitable for transfer of funds from one account to another electronically.

Electronic fund transfer has many advantages over traditional paper currency notes in terms of speed, reliability, cost and convenience. At the same time, it also introduces new kinds of challenges. In this chapter we give a brief introduction of electronic fund transfer systems with focus on offline anonymous untraceable electronic cash also known as e-cash. Electronic fund transfer can be classified in

to online and offline mode on the basis of requirement of connectivity with the central server during the payments.

1.1 Online electronic fund transfer

In online system, at the point of sale, the buyer provides his account details along with the amount of payment to the merchant's point of sale device. The account information of the buyer and the seller are sent to the bank along with payment order. The bank is supposed to have two accounts, one for the buyer and another for the seller. On verifying the credentials of the buyer, the bank transfers the said amount to the seller's account. Usually the account information of the buyer is carried on a plastic card with a magnetic strip.

The key point in this system is that the user does not carry any information regarding money on his card. The money is kept with the bank before and after the payment. If the connection with the bank could not be established then the user can not make the payment.

Credit card and debit card based payments are examples of this type of fund transfer.

1.1.1 Disadvantages of online electronic fund transfer

Some of disadvantages of online electronic fund transfer are the following.

1. **Connection issues**

   The bank is required to be present online at the time of payment even during peak hours which makes the scheme inefficient. Also, if a connection with the server can not be established then the user can not complete the transaction.

2. **User's spending history**

   Banks can easily observe and keep track of all the transactions, including the amount, details of buyer and seller. This information can be used to compromise the privacy of the user. Also, exposing the credit card number to the vendor gives it the ability to impersonate the customer in future purchases.

1.2 Offline electronic fund transfer

In offline electronic fund transfer [20], users carry digital cash, an equivalent to physical money. The digital cash is used to buy commodities and services and is obtained by the user from the bank after undergoing through a well defined withdrawal protocol. At the time of spending digital money, neither the buyer nor the seller is required to establish any direct connection with the bank. After the payment, the seller may transfer the digital money to his account with bank by executing a deposit protocol. The deposit protocol need not be followed immediately after the payment, the merchant can do it at his own convenience.

The key point in this system is the execution of the spending protocol, which is offline without any involvement of the bank. This offline electronic cash is called e-cash.

1.2.1 E-cash

E-cash is money in electronic form, stored in a way similar to any other data, such as audio or video file. The digital representation of e-cash typically contains the coin identity number, blinded user's identity and digital signature of the issuer. A user having the right kind of device can withdraw this money from his bank in a way similar to withdrawal of paper currency. The user can later spend this electronic money for purchasing an item from a merchant. At some later point of time the merchant will have to deposit the values obtained from the user to the bank for its redemption.

E-cash implements an offline system, meaning that at the time when user is spending to the merchant there is no need of bank to come in between. This not only reduces traffic load but also removes the dependence on any communication device during the spending protocol.

1.2.2 Desirable properties of e-cash

The two most important characteristics of any e-cash system are the following.

1. **Off-line**

   Ideally, no connection with the bank need to be made at the time of spending e-cash. In other words, the spending protocol must be committed between the user and the merchant without the presence of any third party.

2. **Anonymity**

   The identity of the user should not be traceable by linking the withdrawal,

spending and deposit protocols together in any way. For a genuine payment, the bank should not be able to figure out who made the payment and to whom. In addition, the bank should not be able to link two different payments to the same user thereby protecting the user's identity.

1.2.3 Challenges in e-cash

Being a data file in the electronic form, it is very easy to reproduce or make duplicate copies of e-cash. Therefore, there are some well known challenges in this domain.

1. **Double spending**

   This is the act of spending the electronic cash more than the allowed number of times (usually one) by the owner of the cash.

2. **Colluding of merchants**

   It may be possible that more than one merchants can combine and form an illegal group to produce fake double spent cash. This is only possible if they get enough information across various payments to do so.

1.3 E-cash models

The implementation of an e-cash system is expected to have the characteristic properties of traditional paper currency note. At the same time it must also satisfy the requirements of electronic payment systems. Various models have been proposed to assimilate the properties of conventional cash in e-cash as described here.

1.3.1 Single-valued coin

In this model the electronic money is realized in the form of virtual coins [16]. All coins are supposed to have same denomination value. Each of these coins are represented by some electronic data. In this model if a coin represents one rupee then a person holding Rs. k will have to store a total of k coins each being unique and differentiable from others.

A coin is indivisible, that is, a single coin having unit value can not be broken down in two part.

A user has to undergo a **withdrawal** protocol (figure 1.1) with the bank to obtain an electronic cash coin. During the payment the user has to undergo a **spending** protocol at the sales point with the merchant. The merchant will submit the coin obtained from the user to bank by executing a **deposit** protocol. The bank will verify the values deposited by the merchant and accept the payment transcript. There are several methods for such protocol. Often, a coin spent once cannot be spent again without going through a deposit or a renewal process.

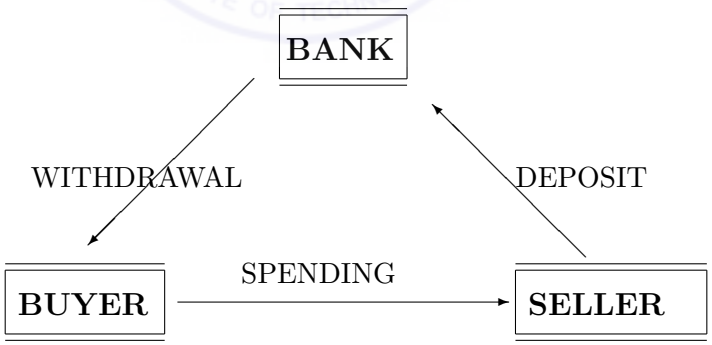

Figure 1.1: Single valued coin based e-cash circulation cycle

1.3.2 Multiple face value coin

Representing different face values on a coin can reduce the space requirement to store a sum of money in the wallet. In this model spending protocol would be fast because it may require fewer number of coin transfers between seller and the buyer. There can be two ways to support this model.

1. **Coins having different face values**

   The model resembles with the real life coinage system where coins are available in different denominations.

   For example, if coins having face values Rs. 1, 2, 5 are supported by the system, then to spend Rs. 3, the user has to spend two coins of face value of Rs. 1 and Rs. 2. On the other hand, this transaction can also be done by giving a coin of Rs. 5 to merchant and taking back a coin of Rs. 2.

   This model introduces new kinds of challenges to deal with like shortage of change and transferability of coin, etc.

2. **Divisible e-cash**

   Divisible e-cash implements the mechanism by which the face value of an e-cash coin can be changed during the spending. It allows to transfer partial face value from one coin to another. By this the e-cash coin can have arbitrary face values.

   In this model when user has a coin of face value Rs. x and he undergoes the spending protocol with a merchant for Rs. y ($y \leq x$) then, after the

completion of spending protocol the user is left with a coin having face value of Rs. $(x - y)$.

1.3.3 Multi-spendable coin

The concept of n-spendable coin implements a coin which has a unit face value but the coin is allowed to participate in spending n number of times (figure 1.2). Therefore, in total the coin will represent a cumulative value of n units. The values representing the coin in its data file remain the same after any spending. It is the responsibility of spender not to spend the coin $(n+1)^{th}$ time.

The protocols ensure that a spender does not spend more than n times without revealing his identity. Until the spender is fair and he undergoes the spending protocol only k number of times $(k \leq n)$ his identity will not be disclosed to the bank, but as he commits $(n+1)^{th}$ spending with the same coin, the bank will get enough information to reveal his identity.

1.3.4 Transferable coin

A transferable coin incorporates the property of re-usability from conventional money. A transferable electronic coin once withdrawn from the bank needs not to be deposited back in the bank after a single payment. The merchant in turn on receiving such a coin can spend the same to another merchant.

In the assumption of bound transferability of a coin, let's define *m-transferable coin* as an electronic coin that when withdrawn through the *bank* by user_1 can go up to user_{m+1} before it must be deposited to the bank (figure 1.3). A user_k

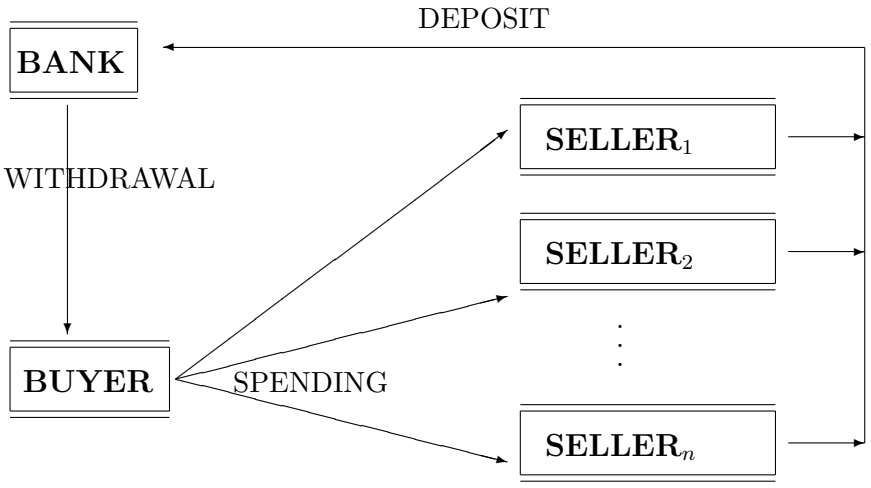

Figure 1.2: Multi-spendable coin withdrawal, spending and deposit protocol cycle

$k < m + 1$ can deposit the coin to *bank* or spend it to user$_{k+1}$, but user$_{m+1}$ must deposit the coin to *bank*.

1.4 Related work

The concept of anonymous offline electronic cash was first introduced by Chaum using blind signatures and cut-and-choose methodology [10, 11]. Later, Chaum, Fiat and Naor [13, 12] demonstrated off-line e-cash construction and laid the foundation for more secure and efficient schemes to follow [18].

Brands presented a more compact way to represent e-cash using restrictive blinding with the help of representation problem in groups [4, 5, 7, 6]. Ferguson used randomized blinding and polynomials to achieve multi-spendability [16]. Camenisch, Hohenberger and Lysyanskaya presented a way to represent e-cash in

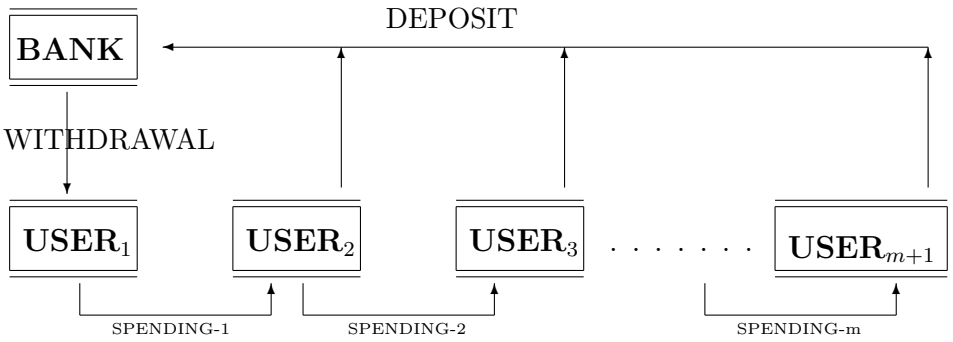

Figure 1.3: m-transferable coin withdrawal, spending and deposit protocol cycle

a compact form using pseudo-random variables [3, 8].

A scheme for transferable e-cash was introduced by Tewari, Mahony and Peirce [23] using observer for coin refreshing. Yahya gave another approach to implement transferable e-cash using secret splitting and a transaction list [27]. Ateniese used group signature scheme with observers to achieve transferability [2]. Other valuable contributions to this area are [21, 22, 25, 26, 24, 14, 9].

1.5 Overview of our work

1.5.1 Motivation

Offline electronic money is likely to be the mode of holding money in future. With the advancement of cyber economy, the use of e-cash is going to increase. This kind of cash should also possess the critical properties of conventional paper currency.

The conventional paper currency notes have the property of offline usability, untraceable and transferability. Out of them, the transferability property appears

to be very critical, it allows a user to withdraw the paper currency money from the bank which he can spend at any shop. The shop keeper being a buyer to his distributor, can spend the same paper currency money obtained from the user to the distributor, and the distributor in turn can further circulate the paper currency money in same way.

This kind of mechanism is also desirable to be incorporated in electronic money. We propose a space efficient e-cash protocol to support m-transferability by developing the concept of coin ownership transfer.

We had also proposed space efficient improvements for single-use, and multi-use e-coin in their existing models to suite for low memory devices.

1.5.2 Organization of this thesis

The rest of this thesis is organized as follows.

In **chapter 2**, we provide the description of a signature based scheme to implement e-cash. Starting with the blind signature based e-cash system which is vulnerable to double spending, a complete scheme free from double spending anomalies is explained here. In **chapter 3**, we explain three things, first the concept of restrictive blinding with the introduction of representation problem in groups. With the use of restrictive blinding protocol, a compact scheme to implement single-use e-cash is also presented. Second, we present the implementation details of n-spendable coin. We explain a polynomial based scheme to achieve n-spendability along with anonymity of the user and a proposal to reduce its space requirements. Third, we focus on transferable e-cash. We present a compact, of-

fline, unlinkable and transferable e-cash scheme. We discuss the present available techniques and advantage of our proposed model for such an e-cash system. In **chapter 4**, we conclude our work and discuss about its future extensions.

Chapter 2

Coin based e-cash

The most accepted model to implement e-cash is based on single denomination units known as e-coin or just the coins. A coin is an indivisible entity, therefore, it is not possible to break the coin in parts for obtaining fractional face values out of it. It is possible to support coins having face values from a pre-defined set, but for our discussion in this chapter we will restrict ourselves to the coins having unit face value.

2.1 E-cash coin life cycle

The life cycle of an e-cash coin consists of the following three phases (figure 2.1).

1. **Withdrawal**

   A coin is obtained by the user through a process of withdrawal from the bank. The user first proves his identity to the bank and then requests for the issuance of coins. The issuance of a coin is committed through the execution of a standard protocol specific to the e-cash model. User obtains

some secure digital value as e-cash coin representation and saves that in his memory device.

2. **Spending**

When the user wants to pay using e-cash coins for a purchase at the sales point, he carries out a spending protocol. Spending is a challenge response based mechanism where the user partially discloses the secret values of the coin to the merchant. These partially disclosed value serve two purposes for the future. Firstly, this makes redemption of a coin possible at the bank and, secondly this may be used as an identity proof of the user in case of a misbehavior like double spending[1] by the user.

3. **Deposit**

A merchant, after successful execution of spending protocol, needs to submit the response values received from the user along with his challenge, to the bank using a deposit protocol. the merchant is however not required to submit these values immediately after the completion of spending. The merchant can do this at his own convenience. After successful completion of the deposit protocol, the e-coin reaches the bank and is destroyed to avoid further use. The bank credits the merchant's account with the destroyed e-coin face value.

[1] see also : section 2.4.4

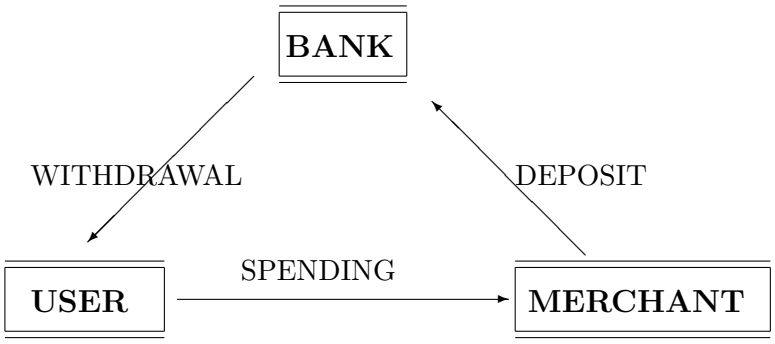

Figure 2.1: E-cash coin life cycle

2.2 Blind signature scheme

Blind signature scheme refers to a method of obtaining digital signature on a value not disclosed to signer. In other words, a user can have a secret message on which he obtains a digital signature from the bank without revealing the message. The scheme that makes it possible is called blind signature scheme. One of the blind signature schemes proposed by D. Chaum [10] that uses RSA algorithm [10] is explained below.

Let's assume that a user wants to obtain digital signature[2] from the bank on a secret message represented as numeric value m. Assume that the bank has RSA public key as (e, n) and RSA private key as (d, n).

The protocol goes as below.

[2]**Digital Signature** on a value m is $(((h(m))^d \bmod n)$ where $h()$ is a one way hash function and (d, n) is RSA private key of the signer.

1. *User* selects a random number $r \in_R \mathbb{Z}_n^*$ [†]. Where n is the modulus of the bank's RSA public key, and $\mathbb{Z}_n^*$ is set of integers modulo n without the identity element.

2. User sends $h(m) \times r^e$ to bank. $h(m)$ is one way hash function value for m. Bank does not gain any knowledge about m or $h(m)$.

3. Bank computes $(h(m).r^e)^d$ using its secret key (d, n) and the received value $h(m).r^e$.

4. Value obtained by the *user* is $(h(m).r^e)^d = (h(m))^d.r^{ed} \equiv (h(m))^d.r$. (Since $ed \equiv 1$). The computed value is send back to the user.

5. *User* computes inverse of r and multiplies this with the obtained response from the *bank* to get $((h(m))^d.r).r^{-1} = (h(m))^d \times rr^{-1} = (h(m))^d \bmod n$. Which is the digital signature on the value m.

2.3 Elementary e-cash protocol

During withdrawal protocol user selects a large random integer A and sends it to the bank for digital signature on it in blinded manner. The random number A and the banks signature on it together make a pair (A, sign(A)) which is called an e-cash coin.

User, during the spending protocol, gives the pair (A, sign(A)) to the merchant. Merchant verifies the digital signature and accepts the payment. At some later

[†]Let S be a set, then $a \in_R S$ represent an element a of set S chosen at random according to the uniform distribution.

point of time, the merchant will produce (A, sign(A)) to the bank for its redemption.

On getting the deposited coin from the merchant the bank will also verify the digital signature of the coin, and upon successful verification accept the coin and release the payment to merchant.

2.3.1 Features

In this e-cash scheme, the coin's identity is randomly picked by the user and is not disclosed to the bank during the withdrawal protocol. Therefore, if the user has two e-cash coins with identity numbers A_1 and A_2 then after spending these coins followed by deposit by merchants, the bank gets both of these coin identity numbers. The bank will not be able to establish that these two coins were withdrawn by the same user. By this, the scheme achieves **unlinkability**.

Due to the same fact, the bank will not be able to guess the identity of the user who generated this coin during withdrawal protocol. By this feature, the scheme achieves the property of **anonymity** and **non-traceability** spending.

2.3.2 Challenges

This approach is very basic and suffers from certain problems. After payment, the original coin representation still remains known to the spender, therefore the user after giving the coin to a merchant, can reproduce and spend the same coin with another merchant. This misbehavior is called **double spending**.

To prevent double spending, the protocol should be designed so that double spending is discoverable. As long as the coin holder does not spend a coin more than once, his identity is kept secret but when he spends it twice or more there would be sufficient information to disclose the identity of the user.

2.4 The protocol

To prevent double spending the representation of a coin requires a change. A protocol was designed by D. Chaum [10] where the coin was represented by t pair of values. Here t is a security parameter. The larger is the t higher is the probability of catching a double spender.

With this modified representation, the protocol changes as follows.

2.4.1 Withdrawal protocol

To get an e-cash coin the *user* sends randomly generated $2t$ pair of values of the following form to the *bank* for its digital signature in blinded manner.

$$\begin{array}{cc} (v_1, & v_1 \oplus I) \\ (v_2, & v_2 \oplus I) \\ (v_3, & v_3 \oplus I) \\ \vdots & \vdots \\ (v_{2t}, & v_{2t} \oplus I) \end{array}$$

Here I is the unique identity of the user that is known to the bank. We can also think of it as a bank account number of the user. The $\oplus$ operator represents an *exclusive-or* operation.

Under blind signature scheme to get digital signature on a value v_i the user sends $h(v_i) \times r_{i1}^e$ to the bank, in the similar way to get signature on $v_i \oplus I$ the user sends $h(v_i \oplus I) \times r_{i2}^e$ to the bank. Where r_{i1} and r_{i2} are additional random numbers selected by the user for blinding. Therefore, during the signature process bank neither knows v_i nor $v_i \oplus I$.

To verify that the user had actually embedded his own identity I in the second part of all the pairs, the bank randomly selects any t pairs out of these $2t$ pairs and ask the user to disclose their respective blinding values (v_i, r_{i1}, r_{i2}). By this the bank does two things.

1. Bank obtains t pairs of v_i and $v_i \oplus I$. Since bank knows I, it can verify that all these pairs are formed in correct manner. This ensures with high probability that other pairs are also been formed in correct manner.

   Bank can choose t pairs out of $2t$ pairs in ${}^{2t}C_t$ ways. Therefore, the probability of guessing in advance the bank's choice is $(1/{}^{2t}C_t)$. Hence, the probability of user getting success in embedding some one else's identity in the e-cash coin without being caught in cut-and-choose protocol is $(1/{}^{2t}C_t)$. This probability reduces exponentially with t.

2. Due to exposure of respective blinding values, t pair of values are destroyed. Therefore, the user is now left with having only t pair of undisclosed values. These undisclosed pair of values together make the e-cash coin.

Let these t pairs of values along with their digital signature be denoted as follows.

$$
\begin{array}{cc}
A_1,\text{sign}(A_1) & A_1 \oplus I,\text{sign}(A_1 \oplus I) \\
A_2,\text{sign}(A_2) & A_2 \oplus I,\text{sign}(A_2 \oplus I) \\
A_3,\text{sign}(A_3) & A_3 \oplus I,\text{sign}(A_3 \oplus I) \\
: & : \\
A_t,\text{sign}(A_t) & A_t \oplus I,\text{sign}(A_t \oplus I)
\end{array}
$$

All these values together make a single coin.

2.4.2 Spending protocol

When a *user* wants to spend the coin and pass it to a merchant, the merchant generates a t-bit random pattern of zeroes and ones.

The spending protocol executes in t steps. At k^{th} step of the spending protocol the merchant sends k^{th} bit of the random pattern as a challenge to the user.

Based on the merchants challenge the user provides response. When the challenge is 0 the user replies A_k and when the challenge is 1 the user replies $A_k \oplus I$ to the merchant. The values are provided along with the digital signature of the bank as shown below.

$$
\text{User's } k^{th} \text{ Response} = \begin{cases} A_k, \text{sign}(A_k) & \text{If } k^{th} \text{ challenge bit is 0} \\ A_k \oplus I, \text{sign}(A_k \oplus I) & \text{If } k^{th} \text{ challenge bit is 1} \end{cases}
$$

For example, consider the following case where merchant generates a random pattern as 001...0.

Coin		Random Bit	Merchant gets
$A_1, \text{sign}(A_1)$	$A_1 \oplus I, \text{sign}(A_1 \oplus I)$	0	$A_1, \text{sign}(A_1)$
$A_2, \text{sign}(A_2)$	$A_2 \oplus I, \text{sign}(A_2 \oplus I)$	0	$A_2, \text{sign}(A_2)$
$A_3, \text{sign}(A_3)$	$A_3 \oplus I, \text{sign}(A_3 \oplus I)$	1	$A_3 \oplus I, \text{sign}(A_3 \oplus I)$
:	:	:	:
$A_t, \text{sign}(A_t)$	$A_t \oplus I, \text{sign}(A_t \oplus I)$	0	$A_t, \text{sign}(A_t)$

2.4.3 Deposit protocol

The *merchant* submits all the user responses {(A_1, sign(A_1)), (A_2, sign(A_2)), ($A_3 \oplus I$, sign($A_3 \oplus I$)), ..., (A_t, sign(A_t))} to the bank. The bank can verify each value by the signature.

2.4.4 Double spending

This scheme ensures that the identity information of a genuine user is kept secret across a single spending. But if the user performs another spending protocol with the same e-cash coin then it provides enough information to reveal his identity.

For example, in our case if user undergoes in another spending protocol with another merchant, he will have to reply to another random t-bit pattern. Consider the following double spending scenario where the second merchant generates a random pattern as 010...0.

Coin		Random Bit	Second merchant gets
$A_1, \text{sign}(A_1)$	$A_1 \oplus I, \text{sign}(A_1 \oplus I)$	0	$A_1, \text{sign}(A_1)$
$A_2, \text{sign}(A_2)$	$A_2 \oplus I, \text{sign}(A_2 \oplus I)$	1	$A_2 \oplus I, \text{sign}(A_2 \oplus I)$
$A_3, \text{sign}(A_3)$	$A_3 \oplus I, \text{sign}(A_3 \oplus I)$	0	$A_3, \text{sign}(A_3)$
:	:	:	:
$A_t, \text{sign}(A_t)$	$A_t \oplus I, \text{sign}(A_t \oplus I)$	0	$A_t, \text{sign}(A_t)$

The probability that both merchants will produce the same random t-bit pattern is very low $(1/2^t)$. Therefore, both merchants will get at least one pair of values different from each other.

When the second merchant submits its acquired values $\{(A_1, \text{sign}(A_1))$, $(A_2 \oplus I, \text{sign}(A_2 \oplus I))$, $(A_3, \text{sign}(A_3))$, ..., $(A_t, \text{sign}(A_t))\}$ to the bank, the bank can immediately recognize the identity of the double spender by exclusive-OR of the different values received (table 2.1).

Value submitted by merchant$_1$	Value submitted by merchant$_2$	taking $\oplus$
A_1	A_1	0
A_2	$A_2 \oplus I$	I
$A_3 \oplus I$	A_3	I
:	:	:
A_t	A_t	0

Table 2.1: Identity exposure of double spender in cut-and-choose based e-coin

Chapter 3

E-cash mechanism

In this thesis we explored three different mechanisms for e-cash.

1. Space efficient single-use e-coin (SESUC).

   SESUC scheme implements an e-coin that is effiecient in terms of storage requirements but can be used only once for spending.

2. Multi-use e-coin (MUC).

   A single MUC can be used by a user as a multi-dimensional e-coin for payment to merchants one unit at a time.

3. Single-use transferable e-coin (SUTC).

   SUTC is a re-usable e-coin that is not required to be deposited to bank after each transaction. A user on receiving an SUTC can use it to buy services or comodeties from another merchant.

3.1 Space efficient single use e-coin (SESUC)

The SESUC scheme is based on the theory of *representation problem in groups* and the method of *restrictive blinding* [4] as described below.

3.1.1 Representation problem in groups

In a group G_q of large prime order q, let us fix a randomly picked generator tuple $(g_1, g_2, ..., g_k)$ where $k \geq 2$ and $g_i \in G_q \setminus \{1\}$ with $g_i \neq g_j$ $(\forall i, j, i \neq j)$. For any $x \in G_q$, a representation of x with respect to $(g_1, g_2, ..., g_k)$ is given by (a_1, a_2, ..., a_k), where $a_i \in \mathbb{Z}_q$ such that $\prod_{i=1}^{k} g_i^{a_i} = x$.

The representation of x however suffers from two major issues as given below.

1. To find the representation of a given number x, with respect to a given randomly chosen generator tuple $(g_1, g_2, ..., g_k)$ is hard.

2. For a given randomly chosen generator tuple $(g_1, g_2, ..., g_k)$, a number x may be obtained by randomly selecting a_1, a_2, ..., a_k and computing $x = \prod_{i=1}^{k} g_i^{a_i}$. However, for this number x it is hard to find another representation. Hence for a given number x, it is hard to find two different representations.

For all $x \in G_q$ and a generator tuple of length k, there are exactly q^{k-1} representations of x. Therefore, probability of guessing a representation is $1/q$ which is low if the value of q is large.

For a group of large prime order, solving the discrete log problem[1] [1] is hard. Therefore, the algorithm that computes a representation of given number x for randomly chosen generator tuple $(g_1, g_2, ..., g_k)$ would also be hard. There exists no polynomial-time algorithm that can output a number and its two different representations with respect to a given generator tuple. This algorithm requires solving of discrete log problem.

3.1.2 Restrictive blinding

It is beneficial to embed the identity of e-cash coin owner in the identity of the coin. This will enable the bank to disclose user's identity in case he gets involved in double spending. Although the coin identities are required to be random, we restricted the coin identities to be only of the form $(I.g_2)^s$ [†], where s is a random number chosen by the user and I is the identity (account number) of the user. This restrictive blinding protocol, ensures that the coin identities are sufficiently random and embed the user identity that can not be revealed.

In other words a user selects $s \in_R \mathbb{Z}_q^*$ and generates a coin identity $A = (I.g_2)^s$. Bank provides blind digital signature on A during the execution of restrictive blinding protocol, after ensuring that it is in proper form, without knowing A.

[1] Let $a \in G_q$ and $b = a^x$ for some integer x. Then given a and b finding x is a discrete log problem. We can also represent the solution as $x = \log_a b$. In particular, we are interested in large multiplicative groups for which the problem of finding discrete log is hard.

[†] g_2 is a constant declared by bank and known to all.

3.1.3 The protocol

3.1.3.1 Setup of system

Bank generates a large prime number q, a random generator tuple $(g,\ g_1,\ g_2)$ where $g, g_1, g_2 \in_R \mathbb{Z}_q^*$, and a secret number $p \in_R \mathbb{Z}_q^*$. Bank also chooses two collision resistant hash function H, H_0 such that

$$H : G_q \times G_q \times G_q \times G_q \times G_q \rightarrow \mathbb{Z}_q^*$$

$$H_0 : G_q \times G_q \times \text{ShopID} \times \text{timeStamp} \rightarrow \mathbb{Z}_q^*$$

Bank publishes the description of G_q, the generator-tuple (g, g_1, g_2), $h(= g^p)$, and the description of H, H_0. The value p is not revealed by the bank.

In our design, the bank follows digital signature scheme as specified by D. Chaum and Pedersen [15]. In this scheme, the signature on a pair of values $(A, B) \in G_q \times G_q$ is a tuple $(z, a, b, r) \in G_q \times G_q \times G_q \times \mathbb{Z}_q$ such that the following two equations are satisfied.

$$g^r = a.h^{H(A,B,z,a,b)} \tag{3.1}$$

$$A^r = b.z^{H(A,B,z,a,b)} \tag{3.2}$$

The method of obtaining the signature is explained in detail under withdrawal protocol (section 3.1.3.3).

For verification of signature, the verifier takes A, B with its signature $\text{sign}(A, B) = (z, a, b, r)$ from user, and two additional parameters g and h from public declara-

tions of bank, to test the satisfiability of equations 3.1 and 3.2. A valid signature satisfies both the equations.

3.1.3.2 Opening of account

To become a user of the e-cash system every buyer and every seller has to register with the bank. Bank provides initial settings in the form of an account number and secret key to the user.

For the registration, user generates $u \in_R \mathbb{Z}_q$, and computes $I = g_1^u$. If $I.g_2 \neq 1$, then user transmits I to the bank, and keeps u a secret. Bank stores I as the account number of user if it is unique. Otherwise a new value of I is generated by the user. Bank also provides g_1^p and g_2^p to the user so that he can compute $z_u = (I.g_2)^p$ ‡.

User		Bank
$u \in_R \mathbb{Z}_q$		
$I = g_1^u$	$\xrightarrow{I}$	unique(I) ?
	$\xleftarrow{g_1^p, g_2^p}$	
$z_u = (g_1^p)^u.g_2^p$		

Finally the user has a secret u, publicly known bank account number I, and z_u. User is also aware of the bank's public parameters (g, g_1, g_2), and h; and public hash functions H and H_0.

‡ $z_u = (I.g_2)^p = ((g_1^u).g_2)^p = (g_1^u)^p.g_2^p = (g_1^p)^u.g_2^p$.

3.1.3.3 Withdrawal protocol

By withdrawal protocol user obtains two numbers, both signed by the bank. First one (A) is used as coin identification number, and the second one (B) is used for blinding purposes during the spending protocol. User knows the representation of both the numbers.

The withdrawal protocol proceeds in the following way.

Step 1 The user initiates the protocol by sending his identity I to the bank.

Step 2 The bank picks a random number $w \in_R \mathbb{Z}_q$, and sends two numbers a_B and b_B to the user, where $a_B = g^w$ and $b_B = (I.g_2)^w$.

Step 3 User picks a random numbers $s \in_R \mathbb{Z}_q^*$, and $x_1, x_2 \in_R \mathbb{Z}_q$ to compute $A = (I.g_2)^s$, $B = g_1^{x_1}.g_2^{x_2}$, and $z = z_u^s$.

By this user generates random coin serial number A, and a blinding number B for which he knows the representation with respect to (g_1, g_2).

User also generates two more random numbers $u, v \in_R \mathbb{Z}_q$ to compute $a = a_B^u.g^v$ and $b = b_B^{s.u}.A^v$. He then computes challenge $c' = H(A, B, z, a, b)$, and sends the blinded challenge $c = c'/u \pmod q$ to the bank.

Simplified values of $a = a_B^u.g^v = (g^w)^u.g^v = g^{w.u+v}$ and $b = b_B^{s.u}.A^v = (I.g_2)^{w.s.u}.(I.g_2)^{s.v} = (I.g_2)^{s.(w.u+v)}$. Since the value of c' is calculated by a hash function therefore, the parameters to the hash function (A, B, z, a, b) can not be modified in arbitrary way.

Step 4 Bank sends the response $r_B = c.p + w$ to the user, and debits the account of user by the face value of the coin.

User accepts r_B if and only if $g^{r_B} = a_B.h^c$ and $(I.g_2)^{r_B} = b_B.z^{c_B}$. If this verification holds, user computes $r = r_B.u + v$.

Where $r = r_B.u+v = (c.p+w).u+v = ((c'/u).p+w).u+v = (c'.p+w.u)+v = c'.p + (w.u + v)$, when $c' = H(A, B, z, a, b)$.

The entire proposal is shown below.

User		Bank
	$\xrightarrow{I}$	$w \in_R \mathbb{Z}_q$
		$a_B = g^w$
		$b_B = (I.g_2)^w$
$s \in_R \mathbb{Z}_q^*$	$\xleftarrow{a_B, b_B}$	
$A = (I.g_2)^s$, $z = z_u^s$		
$x_1, x_2 \in_R \mathbb{Z}_q$		
$B = g_1^{x_1}.g_2^{x_2}$		
$u, v \in_R \mathbb{Z}_q$		
$a = a_B^u.g^v$, $b = b_B^{s.u}.A^v$		
$c' = H(A, B, z, a, b)$		
$c = c'/u \bmod q$	$\xrightarrow{c}$	
		$r_B = c.p + w \pmod q$
$g^{r_B} \stackrel{?}{=} a_B.h^c$, $(I.g_2)^{r_B} \stackrel{?}{=} b_B.{z_u}^c$	$\xleftarrow{r_B}$	
$r = r_B.u + v$		

Finally user gets digital signature [15] of bank on values A and B without disclosing these values to the bank. The signature on these values is given by

$$\text{sign}(A, B) = (z, a, b, r)$$

The verification of this signature holds if $g^r = a.h^{H(A,B,z,a,b)}$ [§], and $A^r = b.z^{H(A,B,z,a,b)}$ [¶].

The representation of e-coin is a tuple $(A, B, \text{sign}(A, B))$, with A being a coin identification number of the form $(I.g_2)^s$, B being a blinding number of the form $(g_1^{x_1}.g_2^{x_2})$ whose representation with respect to g_1, g_2 is known to user, together with a signature on (A, B) which we denote by $\text{sign}(A, B) = (z, a, b, r)$ such that $g^r = a.h^{H(A,B,z,a,b)}$ and $A^r = b.z^{H(A,B,z,a,b)}$.

3.1.3.4 Spending protocol

When user wants to spend his coin to buy services from the merchant, the following protocol is followed.

Step 1 User sends $(A, B, \text{sign}(A, B))$ to the merchant.

Step 2 After verifying the $\text{sign}(A, B)$[§] the merchant computes challenge $x = H_0(A, B, I_{\text{merchant}}, \text{timeStamp})$, where timeStamp is the number representing date and time of the transaction. Merchant sends x to the user.

Step 3 User computes the response $r_1 = usx + x_1$ and $r_2 = sx + x_2$, and sends them back to the merchant.

Step 4 The merchant accepts the payment if and only if $g_1^{r_1}.g_2^{r_2} = A^x.B$.

[§] $g^r = g^{c'.p+(w.u+v)} = g^{c'.p}g^{(w.u+v)} = (g^p)^{c'}.g^{(w.u+v)} = (h)^{c'}.a = a.h^{H(A,B,z,a,b)}$

[¶] $A^r = ((I.g_2)^s)^{(c'.p+(w.u+v))} = (I.g_2)^{s.(c'.p+(w.u+v))} = (I.g_2)^{s.c'.p}.(I.g_2)^{s.(w.u+v)} = (I.g_2)^{s.c'.p}.b = b.((I.g_2)^p)^{s.c'} = b.(z_u)^{s.c'} = b.(z_u^s)^{c'} = b.z^{c'} = b.z^{H(A,B,z,a,b)}$

[§] $\text{sign}(A, B)$ is a tuple (z, a, b, r) its verification holds if $g^r = a.h^{H(A,B,z,a,b)}$, and $A^r = b.(z)^{H(A,B,z,a,b)}$.

User		merchant
	$\xrightarrow[\text{sign}(A,B)]{(A,B)}$	Verify(sign(A, B))
$r_1 = usx + x_1$	$\xleftarrow{x}$	$x = H_0(A, B, I_{\text{merchant}}, \text{timeStamp})$
$r_2 = sx + x_2$		
	$\xrightarrow{r_1, r_2}$	$g_1^{r_1}.g_2^{r_2} \stackrel{?}{=} A^x.B$

If the user during payment protocol can give correct response for a challenge, then he knows a representation of both A and B with respect to (g_1, g_2). Therefore, user can spend a coin if and only if he knows a representation of it.

3.1.3.5 Deposit Protocol

The merchant sends payment transcript to the bank, consisting of A, B, sign(A, B), (r_1, r_2), I_{merchant} and timeStamp of the transaction.

Bank computes x using the identifying number of the shop I_{merchant}, and timeStamp of transaction ($x = H_0(A, B, I_{\text{merchant}}, \text{timeStamp})$). Bank then verifies $g_1^{r_1}.g_2^{r_2} \stackrel{?}{=} A^x.B$ and that sign(A, B) is a signature on (A, B).

If any of the verifications fails, then bank does not accept the payment transcript. Otherwise, Bank searches its deposit database to find out whether A has been stored earlier or not. When A is not found in deposit database the payment is accepted as genuine. Bank then stores (A, x, r_1, r_2) in its database.

3.1.3.6 Double spending

When after a fresh execution of deposit protocol the bank realizes that A is already present in the deposit database the bank retrieves the saved information. Now the bank has at his disposal a pair (x', r_1', r_2') from the new transcript and a pair (x, r_1, r_2) from the saved database.

Since $r_1 = u.s.x + x_1$ and $r_1' = u.s.x' + x_1$ therefore, $r_1 - r_1' = u.s.(x - x')$.

Similarly $r_2 = s.x + x_2$ and $r_2' = s.x' + x_2$ therefore, $r_2 - r_2' = s.(x - x')$.

By use of these two values $(r_1 - r_1')$ and $(r_2 - r_2')$ bank can now compute $(r_1 - r_1')/(r_2 - r_2') = u.s.(x - x')/s.(x - x') = u$. With this the bank can determine the identity of user as follows.

$$\begin{aligned} & g_1^{(r_1 - r_1')/(r_2 - r_2')} \\ \Rightarrow \quad & g_1^u \\ \Rightarrow \quad & I \end{aligned}$$

Bank then searches its account database for this account number, the corresponding account-holder is the double-spender. The number $(r_1 - r_1')/(r_2 - r_2')$ serves as a proof of double spending, it is equal to $\log_{g_1} I$, with I the account number of the double spender.

3.2 Multi-use e-coin (MUC)

By multi-spendability of an e-coin we mean that the e-cash coin is allowed to participate in spending protocol a number of times (say n). This coin takes the storage space for only one coin of a denomination of n. The coin remains same during all instances of spending protocol. The user maintains a separate counter for number of times the coin has been spent so that he does not spend the coin $(n+1)^{\text{th}}$ time. The scheme uses *randomized blinding* [17] protocol and *polynomials* to achieve this feature. The scheme is designed in such a way that the identity of the user is not revealed as long as the coin is spent less than or equal to n times. The spender's identity gets revealed as soon as he spends the coin $(n+1)^{\text{th}}$ time.

We present here the basic scheme for n-spendable coin [17] and then suggest improvement to reduce storage requirement.

3.2.1 Randomized blinding protocol

The randomized blinding protocol is the blind signature protocol[2] that ensures that both parties (user and bank) must use random number as required in the protocol. For example the e-cash coin identity must be a random number, this signature scheme ensures that even if a user does not select random identity number the bank will force it to randomize.

When the user wants digital signature on a random number from the bank, the signature scheme (Table 3.1) proceeds as follows. Let RSA public and private key

[2]The scheme will find digital signature on a value a as $(ag^{f(a)})^d$. where $f()$ is a one way function and d is RSA private key of signer and g is a generator of $\mathbb{Z}_q^*$ publicly declared by the bank.

of bank be (e, n) and (d, n).

Step 1 User selects three random numbers $a_1, b, w \in_R \mathbb{Z}_q^*$ and sends $a_1 b^e g^w$ to the bank where g is a generator of $\mathbb{Z}_q^*$, publicly declared by the bank.

Step 2 Bank provides to the user another random number $a_2 \in_R \mathbb{Z}_q^*$ as response.

Step 3 User replies back to the bank with a number $x = f(a_1 a_2) - w$ where $f()$ is a one way hash function.

Step 4 Bank computes $y = (a_1 b^e g^w . a_2 . g^x)^d$ by multiplying the number received in step 1 with a_2 and g^x followed by raising a power by d to the product. Bank transmits y to the user.

Step 5 User receives $y = (a_1 b^e g^w a_2 g^x)^d = (a_1 a_2 b^e g^w g^{(f(a_1 a_2)-w)})^d = (a_1 a_2 g^{f(a_1 a_2)})^d b^{ed}$ $= (a_1 a_2 g^{f(a_1 a_2)})^d b$ which is equivalent to $(a.g^{f(a)})^d.b$ for $a = a_1.a_2$.

Step 6 Finally the user gets digital signature on the random value a by dividing y with b. Therefore, $\text{sign}(a) = y/b$.

User		Bank
$a_1, b, w \in_R \mathbb{Z}_q^*$		
	$\xrightarrow{a_1 b^e g^w}$	$a_2 \in_R \mathbb{Z}_q^*$
$x = f(a_1.a_2) - w$	$\xleftarrow{a_2}$	
	$\xrightarrow{x}$	$y = (a_1 b^e g^w . a_2 . g^x)^d$
$a = a_1.a_2$	$\xleftarrow{y}$	
$\text{sign}(a) = y/b$		
$(\text{sign}(a))^e \stackrel{?}{=} a.g^{f(a)}$		

Table 3.1: Randomized blind signature

3.2.2 Multi-spendable e-cash protocol

Protocol for withdrawal, spending and deposit proceeds as follows.

3.2.2.1 Withdrawal

For an n-spendable e-cash coin user creates $n+2$ random numbers A, B, C_1, C_2, ..., C_n[†] and he receives $n+1$ RSA signatures as $(h(A)^I h(B))^d$, $(h(A)^{k_1} h(C_1))^d$, $(h(A)^{k_2} h(C_2))^d$, ..., $(h(A)^{k_n} h(C_n))^d$ from the bank. $k_1, k_2, k_3, ..., k_n$ are random numbers known only to the user.

The coin is represented by following set of values

- A, B, C_1, C_2, ..., C_n
- sign(A), sign(B), sign(C_1), sign(C_2), ..., sign(C_n)
- $k_1, k_2, k_3, ..., k_n$
- $(h(A)^I h(B))^d$, $(h(A)^{k_1} h(C_1))^d$, $(h(A)^{k_2} h(C_2))^d$, ..., $(h(A)^{k_n} h(C_n))^d$

3.2.2.2 Spending

The coin can be spent n times. Any of these n allowed spending rounds are carried out in similar manner (Table 3.2) as follows.

Step 1 User sends $A, B, C_1, ..., C_n$ along with their digital signature to the merchant.

[†]Numbers A, B, C_1, C_2, ..., C_n are digitally signed by bank using **randomized blinding protocol**. In the process, user initially selects a random number v_1, which is multiplied by banks response v_2 to create a number $V = v_1.v_2$ on which bank provides its digital signature sign(V).

Step 2 Merchant gives a random challenge x to the user after successful verification of the digital signatures.

Step 3 The user replies back to the merchant with two numbers $r = I + \Sigma k_i x^i$ and $R = (h(A)^I h(B))^d . \prod((h(A)^{k_i} h(C_i))^d)^{x^i}$.

Step 4 The merchant verifies $(R)^e \stackrel{?}{=} h(A)^r h(B) . \prod h(C_i)^{x^i}$. If this verification holds merchant accepts the payment.

The system promises the user to be anonymous up to n spending. Every value provided by the user to satisfy above verification, will be used to form an equation of $k+1$ unknown. For example if during t^{th} spending ($t \leq n$) user on merchant's challenge x_t replies as r_t then $r_t = I + k_1(x_t) + k_1(x_t)^2 + ... + k_1(x_t)^n$ is the equation. Having $k+1$ such equations will make the user disciverable. Thus the, user is forbidden to spend the coin $(k+1)^{\text{th}}$ time or more.

User		Merchant
	$\xrightarrow[\text{sign}(A,B,C_1...C_n)]{A,B,C_1...C_n}$	Verify(sign$(A, B, C_1...C_n)$) $x \in_R \mathbb{Z}_q$
$r = I + \Sigma k_i x^i$ $R = (h(A)^I h(B))^d . \prod((h(A)^{k_i} h(C_i))^d)^{x^i}$	$\xleftarrow{x}$	
	$\xrightarrow{r,R}$	$(R)^e \stackrel{?}{=} h(A)^r h(B) . \prod h(C_i)^{x^i}$

Table 3.2: n-spendable e-cash coin spending protocol

3.2.2.3 Deposit

The merchant deposits $(A, \text{sign}(A), x, r)$ the bank. The bank after successful verification of sign(A) stores (A, x, r) in its database.

3.2.2.4 Unauthorized spending

A user is treated to indulge in unauthorized spending if he participate in spending protocol $(n+1)^{\text{th}}$ time or more by using the same e-cash coin.

As long as the user spends the coin up to n times, bank can not disclose the identity of the spender. This is because $r = I + \Sigma k_i x^i = I + k_1 x^1 + k_2 x^2 + k_3 x^3 + ... + k_n x^n$ where only r and x are known. Using up to n deposited values (A, x_1, r_1), (A, x_2, r_2), ..., (A, x_n, r_n) the *bank* can create following system of equations in $(n+1)$ unknowns $(I, k_1, k_2, ..., k_n)$.

$$
\begin{array}{lllllll}
I & +k_1 x_1^1 & +k_2 x_1^2 & +k_3 x_1^3 & +... & +k_n x_1^n & = r_1 \\
I & +k_1 x_2^1 & +k_2 x_2^2 & +k_3 x_2^3 & +... & +k_n x_2^n & = r_2 \\
I & +k_1 x_3^1 & +k_2 x_3^2 & +k_3 x_3^3 & +... & +k_n x_3^n & = r_3 \\
: & : & : & : & : & : & : \\
I & +k_1 x_n^1 & +k_2 x_n^2 & +k_3 x_n^3 & +... & +k_n x_n^n & = r_n
\end{array}
$$

In order to find the unique solution to get I, the system of equations must be solved with $(n+1)$ independent equations. For a genuine user with having n such equations bank can not disclose the values of $I, k_1, k_2, ..., k_n$. However the system of equations is solvable as soon as another set of value (A, x_{n+1}, r_{n+1}) is known.

Therefore as long as the user spends the coin n times his identity is hidden. But, as soon as he spends $(n+1)^{th}$ time he will provide sufficient number of equations to the bank to disclose his identity by solving the system of linear equations.

3.2.3 Space efficient modified scheme

The space requirement of the representation of the n-spendable e-coin can be reduced by use of *pseudo-random* values for C_i and k_i. We propose a way to construct pseudo-random variable in following way.

User chooses $c \in_R \mathbb{Z}_{t.n}$ here t is any large prime with $t.n < q$ and c is co-prime to $(t.n)$. Let $C = c^t$, then C_i and k_i are formed as follows.

$$C_i = h(C^i) \ / \ 2^{log(q)}, \quad k_i = h(C^i) \bmod 2^{log(q)}$$

User chooses two more random numbers A and B and obtains digital signatures through randomized blinding protocol on A, B, C_1, C_2, ..., C_n. He also receives $n+1$ RSA signatures during the withdrawal protocol as $(h(A)^I h(B))^d$, $(h(A)^{k_1} h(C_1))^d$, $(h(A)^{k_2} h(C_2))^d$, ..., $(h(A)^{k_n} h(C_n))^d$

The coin is represented by following set of values.

- A, B, C
- sign(A), sign(B), sign(C_1), sign(C_2), ..., sign(C_n)
- $(h(A)^I h(B))^d$, $(h(A)^{k_1} h(C_1))^d$, $(h(A)^{k_2} h(C_2))^d$, ..., $(h(A)^{k_n} h(C_n))^d$

The storage requirement of a single e-coin for this new representation requires the storage of $2n + 5$ values. Where as the previous scheme takes the space for $4n + 5$ values. Therefore, we save half the space.

Protocol for opening of an account, withdrawal, spending and deposit protocol proceeds in the same way as described in section 3.2.2.

3.3 Single-use transferable e-coin (SUTC)

In the real world, a person getting money from one person can pass it on to next for buying goods or services. The chain of money handovers starts from a bank (such as Reserve Bank of India) passing through a series of users, and finally ends by reaching back to the bank. Every person in the chain obtains the money from his predecessor[†], and transfers it to his successor by spending. The conventional money being in the form of paper deteriorate with every transaction and is returned to the bank after it becomes unusable over time.

It is desirable for e-cash to mirror all the distinguished properties of conventional paper money. Its transferability turns out to be one of the most useful property to incorporate. This makes the e-cash re-usable. A re-usable electronic coin once withdrawn from the bank need not be deposited back with bank after a single payment. A merchant on receiving such a coin can transfer it to another merchant for buying goods or services.

[†]by earning through spending of predecessor

We had explored the implementation details of one-transferable e-cash coin earlier. These coins have to be returned[‡] to bank after a single spending. Transferability extends this model for re-usability up to a limited extent.

In the assumption of bound transferability of an e-coin, let us define *m-transferable* e-coin as an electronic coin that when withdrawn from the bank by user $\mathcal{U}_1$ can go up to $\mathcal{U}_{m+1}$ before it must be deposited the bank. Any user among $\mathcal{U}_2$, $\mathcal{U}_3$, ..., $\mathcal{U}_k$, $k \leq m$, can deposit it with the bank or transfer it to the next user by spending protocol. The last user in the spending series $\mathcal{U}_{m+1}$ must deposit the coin with the bank for its redemption (figure 3.1).

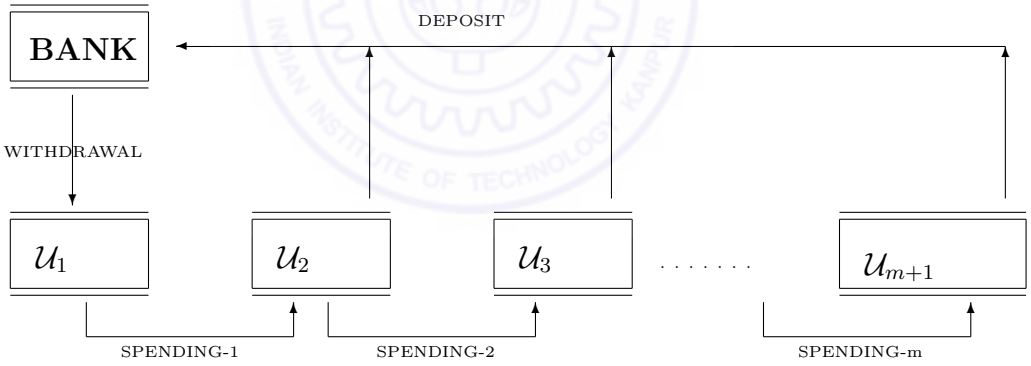

Figure 3.1: Transferable coin withdrawal, spending and deposit protocol cycle

[‡]Coin is returned to bank by the merchant who obtains it after acquiring it from user through spending protocol through user.

3.3.1 Coin representation

In our design, the representation of an m-transferable e-coin contains following information.

1. **Coin identification information**

   Coin identification information is used to differentiate one e-cash coin from another. It serves as the identity number of e-cash coin. The withdrawing user $\mathcal{U}_1$ generates this identity number randomly and gets it digitally signed by the bank. The digital signature of the bank is obtained by executing restrictive blinding protocol to ensure that the bank has no knowledge of the identity number. At the same time the identity of $\mathcal{U}_1$ also get embedded in the identity of the e-coin.

2. **Coin ownership proof**

   E-coin representation contains the identity of its owner in an embedded manner. For the spending purpose, the owner needs to prove that it obtain the coin through genuine channel by executing a challenge response procedure. For this, the user must have enough parameters that enable him to satisfy the spending protocol challenge. To fix these parameters for a particular coin and user, the spender also provides digital signature on them, which we call coin ownership proof. The proof makes the coin spendable only by its owner. A person getting the coin without undergoing through spending protocol can not spend the coin because of the inconsistency of e-cash coin parameters with his credentials.

3. **Spenders history**

   Spenders history is an ordered list of secret sharing information of the users in reusable chain of an e-coin across all spending cycles. This information enables the bank to reveal the identity of a dishonest user if he is involvement in unauthorized spending.

4. **Proof of authenticity**

   Representation of a coin also contains sufficient proofs that its contents are not tampered.

3.3.2 Transferable e-cash coin protocol

The details of transferable e-cash coin withdrawal, spending and deposit protocols are as follows.

3.3.2.1 Initial setup

During initial setup of the system, bank broadcast public parameters of the system, registers every user and provides user-specific settings.

Public parameters Bank generates a random generator-tuple (g, g_1, g_2) where $g, g_1, g_2 \in_R \mathbb{Z}_q^*$, and another random and secret number $p \in_R \mathbb{Z}_q^*$. Bank also chooses a collision-intractable one-way hash function H such that $H : G_q \times G_q \times G_q \times G_q \times G_q \rightarrow \mathbb{Z}_q^*$. Bank publishes the description of G_q, the generator-tuple (g, g_1, g_2), $h(= g^p)$, and the description of H. Bank keeps p a secret.

A signature sign(A, B) [15] of bank on the pair $(A, B) \in G_q \times G_q$ consists of a tuple $(z, a, b, r) \in G_q \times G_q \times G_q \times \mathbb{Z}_q$ such that $g^r = a.h^{H(A,B,z,a,b)}$ and $A^r = b.z^{H(A,B,z,a,b)}$. This is computed by the algorithm as described by D. Chaum [15].

User-specific settings Bank sets following user-specific parameters for every user.

1. **Unique identity**

   Every user in the system gets a unique identity by the bank. The details of identity assignment are as below.

   A user $\mathcal{U}_i$ picks a number $u_i \in_R \mathbb{Z}_q^*$ and sends $I_i = g_1^{u_i}$ to the bank. If I_i is not assigned to anybody, then I_i is registered with bank as the identity of the user$_i$, since u_i is randomly picked by the user and finding discrete *log* of I_i is a hard problem, the bank remains unaware of u_i. User keeps u_i a secret.

   Bank also provides g_1^p and g_2^p to every user so that they can compute $z_i = (g_1^{u_i} g_2)^p$. $(z_i = (g_1^p)^{u_1}.g_2^p)$.

2. **Digital signature on a blinding number**

   For a randomly formed number $B_i = g_1^{v_{i1}} g_2^{v_{i2}}$ where $v_{i1}, v_{i2} \in_R \mathbb{Z}_q$, bank provides digital signature[§] on (I_i, B_i) $\sigma_B(I_i, B_i)$ for the user.

[§] $\sigma_B(m)$ represents the digital signature on a value m done by *bank*.
And $\sigma_{u_k}(m)$ represents the digital signature on a value m done by user$_k$.

Finally the user gets $(u_i, I_i, B_i, \sigma_B(I_i, B_i), z_i)$ from the bank as user-specific parameters.

3.3.2.2 Withdrawal protocol

The withdrawal protocol (Table 3.3) proceeds in following way.

Step 1 The first user initiates the protocol by sending his identity I_1 to the bank.

Step 2 The bank picks a random number $w \in_R \mathbb{Z}_q$, and sends $a = g^w$ and $b = (I_1.g_2)^w$ to the user.

Step 3 User picks a random number $s_1 \in_R \mathbb{Z}_q^*$ to compute $A = (I_1 g_2)^{s_1}$ and $z' = z_1^{s_1}$. User also generates two more random numbers $u, v \in_R \mathbb{Z}_q$ to compute $a' = a^u g^v$ and $b' = b^{s_1 u} A^v$. He then computes challenge $c' = H(A, B, z', a', b')$, and sends the blinded challenge $c = c'/u \bmod q$ to the bank.

Step 4 Bank sends the response $r = cp + w$ to the user, and debits the account of the user.

User accepts r if and only if $g^r = h^c a$ and $(I_1 g_2)^r = z_1^c b$. If this verification holds, user computes $r' = ru + v$.

By this process the user obtains signature on A in a blinded manner. Signature on A is (z', a', b', r') which satisfies two equations $g^{r'} = a' h^{H(A,z',a',b')}$ and $A^{r'} = b' z'^{H(A,z',a',b')}$

Step 5 User sets its correction factor $C_{f1} = 1$ and history parameters $R_{01} = R_{02} = 0$ and send them to the bank for digital signature, this signature need

not to be done in blind manner. Bank supplies digital signature on these values as $\sigma_B(I_1, C_{f1}, R_{01}, R_{02})$.

User		Bank
	$\xrightarrow{I_1}$	$w \in_R \mathbb{Z}_q$ $a = g^w$ $b = (I_1 g_2)^w$
$s_1 \in_R \mathbb{Z}_q^*$	$\xleftarrow{a,b}$	
$A = (I.g_2)^{s_1}$, $z' = z^{s_1}$ $u, v \in_R \mathbb{Z}_q$ $a' = a^u g^v$, $b' = b^{s_1 u} A^v$ $c' = H(A, z', a', b')$		
$c = c'/u \bmod q$	$\xrightarrow{c}$	
		$r = c.p + w \bmod q$
$g^r \stackrel{?}{=} a.h^c$, $(I.g_2)^r \stackrel{?}{=} b.z^c$ $r' = r.u + v$	$\xleftarrow{r}$	
$C_{f1} = 1$ $R_{01} = R_{02} = 0$	$\xrightarrow{C_{f1}, R_{01}, R_{02}}$ $\xleftarrow{\sigma_B(I_1, C_{f1}, R_{01}, R_{02})}$	

Table 3.3: Transferable e-cash withdrawal protocol for first user

User gets following values as an e-cash coin.

- A, $\sigma_B(A)$
- C_{f1}, R_{01}, R_{02}
- R_{11}, R_{12} (both set to 0 by the user)
- $\sigma_B(I_1, C_{f1}, R_{01}, R_{02})$

3.3.2.3 Spending protocol

Transferability imposes a necessary requirement on the spending protocol that the representations of e-cash coin, one with user $\mathcal{U}_{k+1}$ $(k < m)$ after the successful execution of spending protocol, and another with $\mathcal{U}_k$ before the execution of spending protocol, should be equivalent.

Overview: The spending protocol (Table 3.4) can be divided into following steps.

1. Coin identity number verification
2. Proving the coin ownership based on challenge-response.
3. Spender's history verification.
4. Coin ownership transfer and signature.

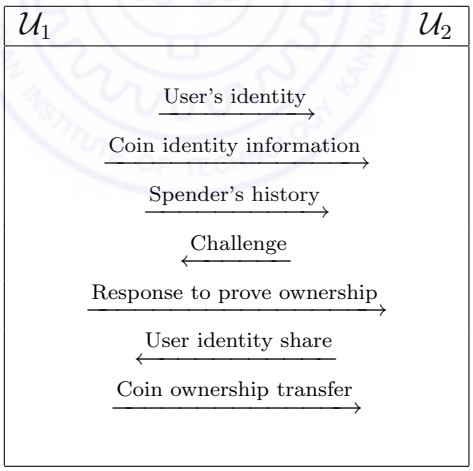

Table 3.4: Overview of spending protocol for a transferable e-cash

The spending protocol is explained here with example.

First spending: Payment by user $\mathcal{U}_1$ to $\mathcal{U}_2$ (Table 3.5) can be explained as follows.

Step 1 The user $\mathcal{U}_1$ initiates the protocol by sending his identity I_1, followed by coin's identity $(A, \sigma_B(A))$, its ownership and acquisition history (C_{f1}, R_{01}, R_{02}, $\sigma_B(I_1, C_{f1}, R_{01}, R_{02})$) to the user $\mathcal{U}_2$.

Step 2 User $\mathcal{U}_2$ after successful verification[3] of digital signature $\sigma_B(A)$ and $\sigma_B(I_1, C_{f1}, R_{01}, R_{02})$ picks a random number $x \in_R \mathbb{Z}_q$ to send a challenge to user $\mathcal{U}_1$.

Step 3 User $\mathcal{U}_1$ respond with two values $r_{11} = u_1 s_1 x + v_{11}$ and $r_{12} = s_1 x + v_{12}$ along with B_1 and $\sigma_B(B_1)$.

Step 4 If user $\mathcal{U}_1$ has ownership of the coin he must be knowing the representation of the coin. User $\mathcal{U}_2$ verifies this knowledge using his response $(AC_{f1})^x.B_1$ and comparing it with $g_1^{r_{11}} g_2^{r_{12}}$. If this verification holds, user$_2$ requests ownership transfer to him by sending $(I_2, I_2^{s_2}, g_2^{s_2})$ to user $\mathcal{U}_1$.

Step 5 User$_1$ provides C_{f2}, R_{11}, R_{12} and its own digital signature on these values $\sigma_{u_1}(I_2, C_{f2}, R_{11}, R_{12})$ as a proof of ownership transfer.

Now user $\mathcal{U}_2$ has following values.

- A, $\sigma_B(A)$
- C_{f2}, R_{11}, R_{12}
- R_{21}, R_{22}

[3]In Table 3.5 and 3.6, vfd(X) represents verification of signature X.

$\mathcal{U}_1$		$\mathcal{U}_2$
	$\xrightarrow{I_1, A, \sigma_B(A), C_{f1}, R_{01}, R_{02}, \sigma_B(I_1, C_{f1}, R_{01}, R_{02})}$	$\text{vfd}(\sigma_B(A))$ $\text{vfd}(\sigma_B(I_1, C_{f1}, R_{01}, R_{02}))$
	$\xleftarrow{x}$	$x \in_R \mathbb{Z}_q$
$r_{11} = u_1 s_1 x + v_{11}$ $r_{12} = s_1 x + v_{12}$	$\xrightarrow{r_{11}, r_{12}, B_1, \sigma_B(B_1)}$	$\text{vfd}(\sigma_B(I_1, B_1))$ $(AC_{f1})^x B_1 \stackrel{?}{=} g_1^{r_{11}} g_2^{r_{12}}$ $s_2 \in_R \mathbb{Z}_q$
$C_{f2} = \frac{I_2^{s_2} g_2^{s_2} C_{f1}}{(I_1 g_2)^{s_1}}$	$\xleftarrow{I_2, I_2^{s_2}, g_2^{s_2}}$	
	$\xrightarrow{C_{f2}, \sigma_{u_1}(I_2, C_{f2}, R_{11}, R_{12})}$	$R_{21} = R_{11}.q + r_{11}$ $R_{22} = R_{12}.q + r_{12}$

Table 3.5: First spending of transferable e-cash

- $\sigma_{u_1}(I_2, C_{f2}, R_{11}, R_{12})$

By this process user $\mathcal{U}_2$ obtains enough parameters to participate in spending protocol with subsequent users.

Subsequent spending: Payment protocol between $\mathcal{U}_k$ and $\mathcal{U}_2$ $(1 \leq k \leq m)$ proceeds in the similar manner to the first payment as follows (Table 3.6).

Step 1 The user $\mathcal{U}_k$ initiates the protocol by sending his identity I_k, followed by coin's identity $(A, \sigma_B(A))$, its ownership and acquisition history (C_{fk}, $R_{(k-1)1}$, $R_{(k-1)2}$, $\sigma_{u_{(k-1)}}(I_k, C_{fk}, R_{(k-1)1}, R_{(k-1)2})$, $r_{(k-1)1}$, $r_{(k-1)2}$) to the user $\mathcal{U}_{k+1}$.

Step 2 User $\mathcal{U}_{k+1}$ after successful verification of digital signatures $\sigma_B(A)$ and $\sigma_{u_{(k-1)}}(I_k, C_{fk}, R_{(k-1)1}, R_{(k-1)2})$ picks a random number $x \in_R \mathbb{Z}_q$ to send a challenge to user $\mathcal{U}_k$.

Step 3 $\mathcal{U}_k$ responds with two values $r_{k1} = u_k s_k x + v_{k1}$ and $r_{k2} = s_k x + v_{k2}$ along with B_k and $\sigma_B(B_k)$.

Step 4 If $\mathcal{U}_k$ has ownership of the coin he must be knowing the representation of the coin. User $\mathcal{U}_{k+1}$ verifies this knowledge using his response $(A.C_{fk})^x.B_k$ and comparing it with $g_1^{r_{k1}}.g_2^{r_{k2}}$. If this verification holds, $\mathcal{U}_{k+1}$ requests ownership transfer to him by sending $(I_{k+1}, I_{k+1}^{s_{k+1}}, g_2^{s_{k+1}})$ to user $\mathcal{U}_k$.

Step 5 $\mathcal{U}_k$ provides C_{fk+1}, R_{k1}, R_{k2} and its own digital signature on these values $\sigma_{u_k}(I_{k+1}, C_{f(k+1)}, R_{k1}, R_{k2})$ as a proof of ownership transfer to $\mathcal{U}_{k+1}$.

Step 6 $\mathcal{U}_{k+1}$ accepts above values if for $R_{k1} = R_{(k-1)1}.q + r_{(k-1)1}$ and $R_{k2} = R_{(k-1)2}.q + r_{(k-1)2}$ the verification of $\sigma_{u_k}(I_{k+1}, C_{f(k+1)}, R_{k1}, R_{k2})$ holds. Finally $\mathcal{U}_{k+1}$ calculates $R_{(k+1)1}$ and $R_{(k+1)2}$.

User $\mathcal{U}_{k+1}$ obtains following values.

- A, $\sigma_B(A)$
- $C_{f(k+1)}, R_{k1}, R_{k2}$
- $R_{(k+1)1}, R_{(k+1)2}$
- $\sigma_{u_1}(I_{k+1}, C_{f(k+1)}, R_{k1}, R_{k2})$

Now he can participate in spending protocol with with subsequent users.

3.3.2.4 Deposit protocol

The last user $\mathcal{U}_{m+1}$ (or any user $\mathcal{U}_k, k \leq m$) deposits $(A, \sigma_B(A), R_1, R_2)$ with the bank along with his own identity.

$\mathcal{U}_k$		$\mathcal{U}_{k+1}$
	$I_k, A, \sigma_B(A), C_{fk}$ $r_{(k-1)1}, r_{(k-1)2}$ $R_{(k-1)1}, R_{(k-1)2}$ $\xrightarrow{\sigma_{u_{(k-1)}}(I_k, C_{fk}, R_{(k-1)1}, R_{(k-1)2})}$	$\text{vfd}(\sigma_B(A))$ $\text{vfd}(\sigma_{u_{(k-1)}}(I_k, C_{fk}, R_{(k-1)1}, R_{(k-1)2}))$
$r_{k1} = u_k s_k x + v_{k1}$ $r_{k2} = s_k x + v_{k2}$	$\xleftarrow{x}$	$x \in_R \mathbb{Z}_q$
	$\xrightarrow{r_{k1}, r_{k2}, B_k, \sigma_B(B_k)}$	$\text{vfd}(\sigma_B(B_k))$ $(AC_{fk})^x B_k \stackrel{?}{=} g_1^{r_{k1}} g_2^{r_{k2}}$
$C_{f(k+1)} = \dfrac{(I_{k+1}^{s_{k+1}} g_2^{s_{k+1}}) C_{fk}}{(I_k g_2)^{s_k}}$	$\xleftarrow{I_{k+1}, I_{k+1}^{s_{k+1}}, g_2^{s_{k+1}}}$	$s_{k+1} \in_R \mathbb{Z}_q$
	$\xrightarrow{C_{f(k+1)}, R_{k1}, R_{k2}}$ $\xrightarrow{\sigma_{u_k}(I_{k+1}, C_{f(k+1)}, R_{k1}, R_{k2})}$	$R_{k1} = R_{(k-1)1}.q + r_{(k-1)1}$ $R_{k2} = R_{(k-1)2}.q + r_{(k-1)2}$ $\text{vfd}(\sigma_{u_k}(I_{k+1}, C_{f(k+1)}, R_{k1}, R_{k2}))$ $R_{(k+1)1} = R_{k1}.q + r_{k1}$ $R_{(k+1)2} = R_{k2}.q + r_{k2}$

Table 3.6: Second and onward spending of transferable e-cash

The bank credits an amount equal to the face value of the coin in the account of user $\mathcal{U}_{m+1}$ upon successful verification of digital signature on A .

3.3.3 Double spending

In the case of malicious double spending (Figure 3.2) by a user (say $\mathcal{U}_k$), the fraud will be detected by the bank during deposit when the duplicate coin is deposited. Let us say that two users $\mathcal{U}_p$ and $\mathcal{U}_q$ wish to deposit $(A, \sigma_B(A), R_{1p}, R_{2p})$ and $(A, \sigma_B(A), R_{1q}, R_{2q})$ with the bank.

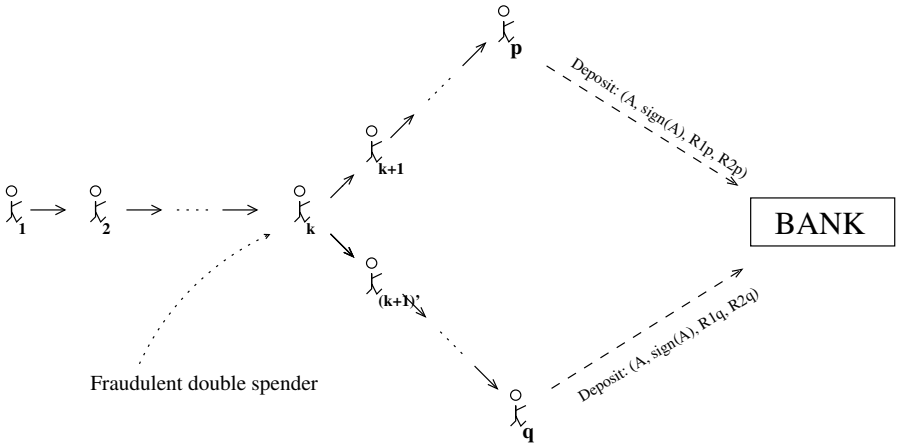

Figure 3.2: Double spending scenario in transferable e-cash

Since these deposits happen at two different time, one user (say $\mathcal{U}_p$) shall successfully deposit the coin with bank. When $\mathcal{U}_q$ reaches the bank to deposit his coin, the bank can unfold all values of (r_{11}, r_{21}, r_{31}, ..., r_{m1}) and (r_{12}, r_{22}, r_{32}, ..., r_{m2}) by using modulo division operation on R_{1p} and R_{2p}. In the same way using modulo division operation on R_{1q} and R_{2q} bank can also unfold values (r'_{11}, r'_{21}, r'_{31}, ..., r'_{m1}) and (r'_{12}, r'_{22}, r'_{32}, ..., r'_{m2}).

A simple comparison will reveal the position and secret share of the double spender. Out of these two sequences (r_{11}, r_{21}, r_{31}, ..., r_{m1}) and (r'_{11}, r'_{21}, r'_{31}, ..., r'_{m1}) bank will observe that the sub-sequences (r_{11}, r_{21}, r_{31}, ..., $r_{(k-1)1}$) and (r'_{11}, r'_{21}, r'_{31}, ..., $r'_{(k-1)1}$) same.

Bank after finding the value of k gets two set of values (r_{k1}, r_{k2}) and (r'_{k1}, r'_{k2}). These values will reveal identity of double spender as follows.

$$I_k = g_1^{(r_{k1}-r'_{k1})/(r_{k2}-r'_{k2})}.$$

3.3.4 Discussion

3.3.4.1 Spender's anonymity

It may be noted that even though the bank can know r_{i1}, r_{i2} for each user $\mathcal{U}_i$, $i \leq m$ these values are random and can not reveal the identity of the user till a fraudulent user $\mathcal{U}_k$ double spends the coin and thereby provide enough information to reveal his identity. Therefore, a genuine spender remains anonymous.

3.3.4.2 Double spender tracing

A fraudulent user $\mathcal{U}_k$ who replies to two or more challenges, would provide more than one pair of values (r_{k1}, r_{k2}) - any two of them can be used to disclose his identity as $I = g_1^{(r_{k1}-r'_{k1})/(r_{k2}-r'_{k2})}$.

3.3.4.3 Ownership transfer

A user owning the coin carries the representation of e-coin which is not known to any one else. For the ownership transfer between $\mathcal{U}_i$ and $\mathcal{U}_{i+1}$ the secret part of e-coin representation s_{i+1} for $\mathcal{U}_{i+1}$ is randomly generated by the $\mathcal{U}_{i+1}$ and $g_2^{s_{i+1}}$ is send to the $\mathcal{U}_i$.

Since finding discrete log is hard therefore, $\mathcal{U}_i$ will not be able to find s_{i+1} but, he can calculate $C_{f(i+1)}$ by using his secret value s_i and received value $g_2^{s_{i+1}}$. $\mathcal{U}_i$ provides $C_{f(i+1)}$ and his digital signature on it to $\mathcal{U}_{i+1}$.

Coin ownership transfer requires the secret values of the e-coin on both sides ($\mathcal{U}_i$ and $\mathcal{U}_{i+1}$) therefore, ownership is a two party mutual activity which can not be carried out by a single individual.

3.3.4.4 History

Spenders history is an ordered list of secret sharing information of users in reusable chain of an e-coin across all the spending. Spenders history is constructed in incremental way as the e-coin passes from one user to another. This information enables the bank to reveal the identity of a dishonest user involvement in unauthorized spending.

Our scheme stores spending history in two numbers R_1 and R_2. For a m-transferable e-coin R_1 and R_2 are required to be of $m.\log(q)$ bit in size.

To protect spending history from being tampered by its owner, spender provides digital signature on it linked with e-coin representation. Therefore any tampering makes the coin unusable for further spending. Spending protocol is designed in such a way that every user not only verifies the digital signature but also its incremental construction by the predecessor.

Chapter 4

Conclusions and future work

4.1 Conclusion

The practical way of making payment requires the transfer of physical cash from the buyer to seller in lieu of some goods or services. The sense of belonging of cash with the seller gives him the satisfaction of completing the transaction. Seller now can act as buyer to the next seller, where he pays the same cash to purchase some commodities.

The paper used cash system has the feature of belongingness and transferability. Any cash system without support for both of these features would be of limited use.

The electronic cash system, e-cash uses digital data for representation and has the feature of belongingness by nature. However transferability is a critical issue. In current transferability solutions of e-cash a person is allowed to spend the cash received from a payer to a merchant, but before spending he has to contact an

observer to refresh the coin. This kind of system seems to be as inconvenient as depositing the e-coin with the bank for its redemption and then requesting for issuance of a fresh e-coin.

In our work, we had proposed a scheme for bounded transferable e-coin where coin refreshing with either a bank or observer is not required. Our scheme is based upon the concept of spending ownership transfer. Representation of transferable e-cash involves parameters dependent upon the identity of its owner and secret values known only to its owner. When the coin is transferred to a merchant he also need to have similar representation. Protocol ensures that a user having knowledge of coin representation can only remodel the coins parameters to become spendable by its successive owner.

We also proposed modifications in existing e-cash models to reduce their space requirements for suitability to low memory devices like PDA or smart-cards. For multi-spendable e-coin our proposal takes the storage space of $2n + 5$ values as compared to $4n + 5$ values in the original protocol.

4.2 Future work

Our work can be extended in several ways. While we have considered the ownership transfers, the coin still remains of the same face value. The following extensions are possible in future.

1. **Multiple face valued e-coin**

   Our solution does not addresses the representation of multiple face values

on e-coin. Incorporating this feature will add more value to e-cash protocol. One of our early suggestion in this direction is to use multiple powers on the coin identifier. Let A be the identification number of a coin having unit face value, then A^v represents a coin of face value v. The challenge here is how to protect value of v from fraudulus modifications.

2. **Unbounded transferability**

   Limited transferability restricts the use of e-coin after a fixed number of times. Our work can be extended to achieve unbounded transferability where the coin keeps circulating indefinitely.

3. **Hiding the identity of own coin**

   It is possible to identify the coin by a user who had seen it before. In other words if an e-coin reaches again to the same user, he will be able to conclude about his previous ownership with the coin. This provides the user extra knowledge about the coin, undesirable for some usage. Our work can be extended to incorporate the hiding of this identification.

Bibliography

[1] L. Adleman. A subexponential algorithm for the discrete logarithm problem with applications to cryptography. In *Foundations of Computer Science, 1979., 20th Annual Symposium on*, pages 55–60, 1979.

[2] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. *Lecture Notes in Computer Science*, pages 255–270, 2000.

[3] M.H. Au, W. Susilo, and Y. Mu. Practical compact e-cash. *LECTURE NOTES IN COMPUTER SCIENCE*, 4586:431, 2007.

[4] S. Brands. An efficient off-line electronic cash system based on the representation problem. *Centrum voor Wiskunde en Informatica, Computer Science/Departement of Algorithmics and Architecture, Report CS- R*, 1993.

[5] S. Brands. Untraceable off-line cash in wallets with observers. *Lecture Notes in Computer Science*, 773:302–318, 1994.

[6] S. Brands. Electronic cash on the Internet. In *Network and Distributed System Security, 1995., Proceedings of the Symposium on*, pages 64–84, 1995.

[7] S. Brands. *Restrictive blinding of secret-key certificates*. Springer, 1995.

[8] J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Compact e-cash. In *Eurocrypt*, pages 302–321. Springer, 2005.

[9] J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. *Lecture Notes in computer science*, 1294:410–424, 1997.

[10] D. Chaum. Blind signatures for untraceable payments. In *Advances in Cryptology: Proceedings of Crypto*, volume 82, pages 199–203, 1983.

[11] D. Chaum. Achieving electronic privacy. *High Noon on the Electronic Frontier: Conceptual Issues in Cyberspace*, page 225, 1996.

[12] D. Chaum, A. Fiat, and M. Naor. "Untraceable Electronic Cash", Crypto'88, LNCS 403.

[13] D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. In *Proceedings of CRYPTO*, volume 88, pages 319–227. Citeseer, 1988.

[14] D. Chaum and T.P. Pedersen. Transferred cash grows in size. *Lecture Notes in Computer Science*, pages 390–390, 1993.

[15] D. Chaum and T.P. Pedersen. Wallet databases with observers. *Lecture Notes in Computer Science*, pages 89–89, 1993.

[16] N. Ferguson. Single term on-line coins. In *Proceedings of Eurocrypt*, volume 93, pages 318–328. Citeseer.

[17] N. Ferguson. Extensions of single-term coins. *Lecture Notes in Computer Science*, pages 292–292, 1994.

[18] R. Hirschfeld. Making electronic refunds safer. *LECTURE NOTES IN COMPUTER SCIENCE*, pages 106–106, 1993.

[19] L. Law, S. Sabett, and J. Solinas. How to make a mint: the cryptography of anonymous electronic cash. *American University Law Review*, 46:1131, 1996.

[20] C.G. Ma and Y.X. Yang. Transferable off-line electronic cash. *Jisuanji Xuebao(Chin. J. Comput.)*, 28(3):301–308, 2005.

[21] C.G. Ma and Y.X. Yang. Transferable off-line electronic cash. *Jisuanji Xuebao(Chin. J. Comput.)*, 28(3):301–308, 2005.

[22] B. Schoenmakers. Basic security of the ecash payment system. *Computer Security and Industrial Cryptography: State of the Art and Evolution*, pages 342–356, 1998.

[23] H. Tewari, D. OMahony, and M. Peirce. Reusable off-line electronic cash using secret splitting. Technical report, Citeseer.

[24] V. Varadharajan, K.Q. Nguyen, and Y. Mu. On the design of efficient RSA-based off-line electronic cash schemes. *Theoretical Computer Science*, 226(1-2):173–184, 1999.

[25] H. Wang and Y. Zhang. A protocol for untraceable electronic cash. *Lecture notes in computer science*, pages 189–200, 2000.

[26] H. Wang, Y. Zhang, and J. Cao. An electronic cash scheme and its management. *Concurrent Engineering*, 12(3):247, 2004.

[27] M. Yahya. General Purpose E-commerce System. 2007.