ABSTRACT

Title of thesis: VARIOUS ASPECTS OF DIGITAL CASH

Haejung Park, Master of Arts, 2008

Thesis directed by: Professor Lawrence C. Washington
Department of Mathematics

In this thesis, we study various aspects of digital cash systems. In particular, we concentrate on two schemes, one due to Okamoto-Ohta and the other due to Eng-Okamoto. The main part of the thesis is devoted to providing more detailed explanations of the issues of divisibility and prevention of double-spending. We then study a few additional systems to explain other aspects of electronic cash.

# VARIOUS ASPECTS OF DIGITAL CASH

by

Haejung Park

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Arts
2008

Advisory Committee:

Professor Lawrence C. Washington, Chairman/Advisor
Professor Thomas Haines
Professor Harry Tamvakis

# ACKNOWLEDGEMENTS

My utmost gratitude goes to my thesis advisor, Dr. Larry Washington, for his kindness, encouragement, and most of all, his patience. It has been the highlight of my time in Maryland to study with him.

I would like to express my gratitude to all the professors who have taught me at the University of Maryland. I also would like to thank the Department of Mathematics and its staff for their support and for a generous fellowship.

My heartfelt gratitude goes to my undergraduate math advisor, Professor Adam Fieldsteel, for helping me find the true joy and beauty of studying mathematics. I am deeply indebted to him for his teaching. I also would like to thank two of my undergraduate math professors, Professor Petra Bonfert-Taylor and Professor Edward Taylor, for their teaching, encouragement, and friendship over the years.

I express my special thanks to Mr. Houghton Freeman and Mrs. Doreen Freeman for giving me the opportunity to study at Wesleyan University. My deep gratitude also goes to the Kilbys for years of love and support, and to my friend, Nancy, for helping me get through tough times.

Finally, I would like to give my thanks to my parents, to whom I dedicate this thesis.

TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# Chapter 1

# Introduction

Electronic Cash is a form of payment that is exchanged electronically. Customers withdraw electronic coins from a bank and pay merchants with them. The Merchants then deposit the coins to the Bank. This system involves computer networks over which the payment is made. In the field of cryptography, the topic has gained much attention over the past few decades. However, the use is still relatively limited and small in scale.

One example is the Octopus card system in Hong Kong. The system began as a public transit payment scheme and is now used more generally as an electronic cash system. Singapore also has the same type of public transportation card. For general purposes, the Netherlands has implemented an electronic cash system, Chipknip.

There are many important features to an ideal electronic cash system. One of them is providing anonymity to the customers as in the case of real cash. Customers should be able to engage in transactions without having their identities revealed. It is also critical that two transactions made by the same person are not traceable by either the bank or the merchant.

Another important aspect of electronic cash is prevention of duplication. Unlike real cash, it is possible to make identical duplicates of electronic coins. In fact, this is relatively easy, whereas duplicating real cash is quite difficult. This creates the problem of double-spending in electronic schemes. An ideal electronic cash system, therefore, should secure the anonymity of honest customers, while revealing the identity of cheating customers and merchants.

Another major concern is divisibility of the electronic coins. Divisible electronic cash schemes allow a customer to withdraw a coin of worth $\$2^{\ell}$, to divide the value, and then to spend the pieces. We will examine this feature in detail shortly.

Overall, there are many aspects to observe in the electronic cash schemes. Different authors have proposed various electronic cash systems, claiming to provide certain sets of features. In this thesis, we concentrate on studying the main two aspects, divisibility and prevention of double-spending. A major part of the thesis will be devoted to adding explicit details to the arguments that are only sketched in Okamoto-Ohta and Eng-Okamoto. We will also look at some other papers that use untraceability, unlinkability, transferability, and zero-knowledge proofs.

In all the cash systems presented in this thesis, we have a set of participants. Namely, the Bank (B), the Customer (U), and the Merchant (M). Different protocols are presented for each system, but they serve the same set of purposes.

An Ideal Cash System should have:

1. **Independence:** The security of an electronic coin should not depend on physical properties of the coin. It should be possible to transfer it electronically over a secure network.

2. **Security:** One should not be able to copy and forge the cash. It is fairly easy to make exact duplicates of an electronic document. This exposes the electronic cash systems to the danger of double-spending. Therefore, making the system secure is directly related to prevention of copying the coins.

3. **Untraceability (privacy):** One cannot trace the relationship between the user and his purchases. In the real cash system, when a customer spends a bill, neither the bank nor the merchant can trace the identity of the customer. We want to replicate this feature in the electronic cash system in such a way that the customer's identity is revealed only if he cheats.

4. **Off-line payment:** The merchant doesn't need to be linked to the Bank before accepting a coin from the Customer. $M$ can simply collect coins from different customers and deposit them later with $B$, when he brings them to exchange e-cash for cash. Even though he accepted the coins from customers without verifying with the Bank, he is ensured that the coins will be accepted by the Bank if legitimate, or the cheating Customer will be identified. Off-line schemes should also protect against cheating Merchants who try to deposit a coin twice.

5. **Transferability:** Once an electronic coin is issued to a Customer, he should be able to transfer all or a portion of the coin's value to another customer. The system must protect both the first and the second customers from cheating against each other.

6. **Divisibility:** The Customer should be allowed to divide the value of an electronic coin in any number of pieces he wants and to spend it one piece

at a time.

7. **Unlinkability:** If the Customer $U$ spends a portion of an electronic coin with one Merchant, and then spends another portion of the same coin with a different Merchant, the two transactions should not enable the Bank to tell whether the two were made by the same person.

Okamoto-Ohta [9] lists the first six requirements. The unlinkability requirement has been studied more recently. Note that credit cards do not satisfy (3). Prepaid cards do not satisfy (1) and (7).

A company, DigiCash, is creating an electronic cash system that enables issuers to sell electronic coins at some value to individuals, who store them in their own computers. However, it is not clear how this system could guarantee untraceability since a third party records all transactions.

Digital cash provides many obvious benefits. But the solutions so far proposed still leave room for further developments and improvement.

# Chapter 2

# Universal Electronic Cash: Okamoto-Ohta

## 2.1 Introduction

The paper [9] proposes the first ideal untraceable electronic cash system. This system satisfies all six criteria that were mentioned earlier. The main feature of the new system is the divisibility of the customer's coin into many pieces in any way he wants. The security of the system lies in the difficulty of factoring.

In using a cut-and-choose methodology, Okamoto and Ohta use a technique involving square roots modulo $N$, where $N$ is a Williams integer, and a binary tree.

## 2.2 Preliminaries

**Definition 1.** *$N$ is called a* **Blum integer** *if $N = pq$, where $p$ and $q$ are primes, $p \equiv 3 \pmod 4$ and $q \equiv 3 \pmod 4$. $N$ is called a* **Williams integer** *if $N = pq$, where $p$ and $q$ are primes, $p \equiv 3 \pmod 8$ and $q \equiv 7 \pmod 8$.*

Observe that a Williams integer is also a Blum integer.

**Definition 2.** *Let $p$ be an odd prime and let $a \not\equiv 0 \pmod{p}$. Define the Legendre symbol $(a/p) = +1$, if $x^2 \equiv a \pmod{p}$ has a solution, $-1$ if $x^2 \equiv a \pmod{p}$ has no solution.*

Some properties of the Legendre symbol are:

**Proposition 1.** *Let $p$ be an odd prime.*

1. *If $a \equiv b \not\equiv 0 \pmod{p}$, then $(a/p) = (b/p)$.*
2. *If $a \not\equiv 0 \pmod{p}$, then $(a/p) \equiv a^{(p-1)/2} \pmod{p}$.*
3. *If $ab \not\equiv 0 \pmod{p}$, then $(ab/p) = (a/p)(b/p)$.*
4. *$(-1/p) = (-1)^{(p-1)/2}$.*

We are now ready to define Jacobi symbol, which extends the Legendre symbol from primes $p$ to composite odd integers $n$.

**Definition 3.** *Let $n$ be an odd positive integer and let $a$ be a nonzero integer with $\gcd(a, n) = 1$. Let $n = p_1^{b_1} p_2^{b_2} \ldots p_r^{b_r}$ be the prime factorization of $n$. Then $(a/n) = (a/p_1)^{b_1}(a/p_2)^{b_2} \ldots (a/p_r)^{b_r}$.*

If $N = pq$ (where $p, q$ are primes), we can classify the elements of $\mathbf{Z}_N^*$ into four classes:

$$\mathbf{Z}_{(1,1)} = \{x \in \mathbf{Z}_N^* \mid (x/p) = 1, (x/q) = 1\}$$

$$\mathbf{Z}_{(1,-1)} = \{x \in \mathbf{Z}_N^* \mid (x/p) = 1, (x/q) = -1\}$$

$$\mathbf{Z}_{(-1,1)} = \{x \in \mathbf{Z}_N^* \mid (x/p) = -1, (x/q) = 1\}$$

$$\mathbf{Z}_{(-1,-1)} = \{x \in \mathbf{Z}_N^* \mid (x/p) = -1, (x/q) = -1\}$$

**Proposition 2.** *Let $N$ be a Blum integer, and let $x$ be an element of $\mathbf{Z}_{(1,1)}$. Then, for any integer $t \geq 1$, there are four values $y_1, y_2, y_3, y_4$ such that $y_i^{2^t} \equiv x \pmod{N}$ and such that $y_1 \in \mathbf{Z}_{(1,1)}$, $y_2 \in \mathbf{Z}_{(1,-1)}$, $y_3 \in \mathbf{Z}_{(-1,1)}$, $y_4 \in \mathbf{Z}_{(-1,-1)}$. In addition, $y_1 \equiv -y_4$ and $y_2 \equiv -y_3$. Also, $(y_1/N) = (y_4/N) = 1$ and $(y_2/N) = (y_3/N) = -1$.*

The proposition is proved using by working mod $p$ and $q$ separately, then using the Chinese remainder theorem.

A consequence of this proposition is that four values of a $2^t$-th root $y$ of $x$ can be uniquely specified by two bits of information; the value of $(y/N)$ and whether $0 < y < N/2$ or not.

**Proposition 3.** *Let $N = pq$ be a Williams integer. Then, for any $x \in \mathbf{Z}_N^*$, there is a unique $a \in \{\pm 1, \pm 2\}$ such that $ax \in \mathbf{Z}_{(1,1)}$.*

This follows immediately from the facts

$$(-1/p) = (-1/q) = -1, \quad (2/p) = -1, \quad (2/q) = 1.$$

**Definition 4.** *Let $N$ be a Williams integer and let $x \in \mathbf{Z}_{(1,1)}$. Let $t \geq 1$. Let*

$$[x^{1/2^t}]_{QR}$$

*denote the value of $y \in \mathbf{Z}_{(1,1)}$ such that $y^{2^t} \equiv x \pmod{N}$. Let*

$$[x^{1/2^t}]_1$$

*denote the value of $y \in \mathbf{Z}_N^*$ such that $(y/N) = 1$ and $0 < y < N/2$. Let*

$$[x^{1/2^t}]_{-1}$$

*denote the value of $y \in \mathbf{Z}_N^*$ such that $(y/N) = -1$ and $0 < y < N/2$. For $z \in \mathbf{Z}_N^*$, let*

$$\langle z \rangle_{QR} = dz \mod N,$$

*where* $d \in \{\pm 1, \pm 2\}$ *is chosen so that* $dz \in \mathbf{Z}_{(1,1)}$*, and let*

$$\langle z \rangle_1 = d'z \mod N,$$

*where* $d' \in \{1, 2\}$ *is chosen so that* $(d'z/N) = 1$.

The following are the main keys to catching double spenders in the Okamoto-Ohta scheme.

**Factorization principle.** Let $n$ be an integer and suppose there exist integers $x$ and $y$ with $x^2 \equiv y^2 \pmod{n}$, but $x \not\equiv \pm y \pmod{n}$. Then $n$ is composite, and $\gcd(x - y, n)$ gives a nontrivial factor of $n$.

For a proof, see [12].

**Proposition 4.** *Let* $N$ *be a Williams integer and let* $x, y \in \mathbf{Z}_N^*$*. If* $(x/N) \neq (y/N)$ *then* $x \not\equiv \pm y \pmod{N}$.

*Proof.* Suppose $x \equiv \pm y \pmod{N}$. Then

$$(x/N) = (\pm y/N) = (\pm 1/N)(y/N) = (y/N),$$

since $(-1/N) = (-1/p)(-1/q) = (-1)(-1) = 1$. □

### 2.2.1 The Binary Tree

A binary tree allows the bill $C$ to be subdivided into pieces such that each subdivided piece is worth any desired value less than C and the total value of all pieces is equal to C. The table is a tree of $t$ levels, in which each node has two offspring nodes. The root node is located at the top of the tree. At the $i$th level, there are $2^{i-1}$ nodes. To demonstrate how the tree works in this cash system, Okamoto uses the tree with three levels, and the value of the issued bill is \$100.

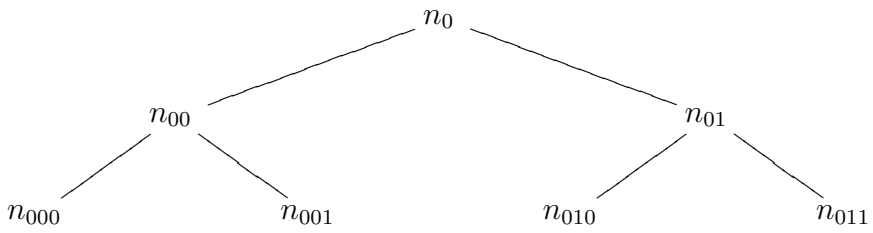


Figure 2.1: The nodes

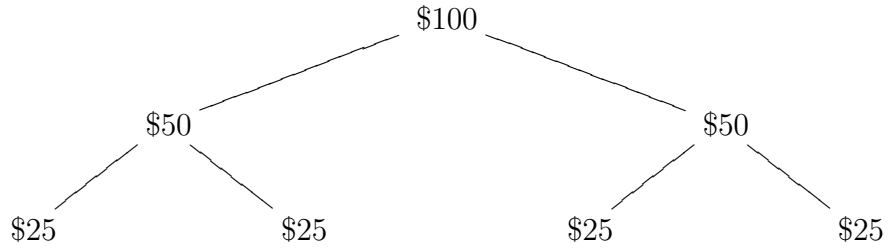


Figure 2.2: Cash values for nodes

The nodes of the $i$th level correspond to $\$100/2^{i-1}$, so the amounts that can be spent are \$25, \$50, \$75, and \$100.

Suppose Alice uses \$75 first and then uses \$25. For spending the \$75, she can use node $n_{00}$ worth \$50 and node $n_{010}$ worth \$25. She then uses node $n_{011}$ to spend the remaining \$25.

The restrictions to the bill spending are as follows:

1. The value of a node $n$ is the total of the values of the nodes that are the direct offsprings of $n$.

2. When a node is spent, all descendant nodes and all ancestor nodes of this node cannot be used.

3. No node can be used more than once.

## 2.3 The Electronic Cash Scheme

### 2.3.1 Initialization

The bank ($B$) generates RSA keys

$$(e_B, n_B; d_B), (e'_B, n'_B; d'_B), \ldots,$$

where $(e_B, n_B), (e'_B, n'_B), \ldots$ are public keys and $d_B, d'_B, \ldots$ are the corresponding secret keys. $(e_B, n_B)$ corresponds to the electronic license that $B$ issues, and $(e'_B, n'_B), \ldots$ correspond to the values of the electronic bills that $B$ issues. Each possible value of a coin has a different signature key. A security parameter $K$ (for example, $K = 40$) and cryptographic hash functions, $f_\Gamma, f_B, f_\Omega, g$ are set by the Bank. The hash functions are used to generate the numbers associated to the binary tree.

### 2.3.2 Opening an Account and Obtaining an Electronic License

The Customer $U$ has identification number $ID_U$. The Customer generates an RSA key $(e_U, n_U; d_U)$, and publishes it. When the Customer opens an account at the bank $B$, the bank issues an electronic license $L$. The Customer $U$ obtains $L$ while opening the account by the following protocol.

1. For $1 \leq i \leq K$, Customer $U$ chooses a random value $a_i$ and a Williams integer $N_i = p_i q_i$, where $p_i \equiv 3 \pmod 8$ and $q_i \equiv 7 \pmod 8$.

2. $U$ chooses random integers $r_i \in \mathbf{Z}_{n_B}$ and forms the numbers

$$S_i = ID_U \| a_i \| (g(ID_U \| a_i))^{d_U} \mod n_U$$

$$S_{1,i} \| S_{2,i} = S_i$$

$$I_{1,i} \equiv S_{1,i}^2, \quad I_{2,i} \equiv S_{2,i}^2 \mod N_i$$

$$I_i = I_{1,i} \| I_{2,i}$$

$$W_i = r_i^{e_B} g(I_i \| N_i) \mod n_B,$$

   where $\|$ denotes concatenation. $U$ sends the numbers $W_i$ to the bank $B$.

3. $B$ chooses some random set of $K/2$ indices $i_1, \ldots, i_{K/2}$, with $1 \leq i_j \leq K$, and sends these to $U$.

4. $U$ sends the following to $B$:

$$a_i, \quad p_i, \quad q_i, \quad (g(ID_U \| a_i))^{d_U} \mod n_U, \quad ID_U, \quad r_i$$

   for all $i \notin \{i_j \mid 1 \leq j \leq K/2\}$. $B$ checks that they are valid.

5. $B$ sends

$$\left(\prod_{i=1}^{K/2} W_i\right)^{d_B} \mod n_B$$

to $U$.

6. $U$ computes

$$L = \left(\prod_{j=1}^{K/2} g(I_{i_j} \| N_i)\right)^{d_B} \mod n_B$$

by dividing by $\prod r_{i_j}$.

It is important to observe that $B$ has signed the numbers $g(I_{i_j} \| N_{i_j})$ without knowing their values. $U$ had disguised these values with $r_i^{e_B}$. This is called a blind signature and is a key point in protecting anonymity of $U$. If $B$ knows the values of $g(I_{i_j} \| N_{i_j})$, then $B$ can identify $U$ as follows: When the coin is spent, $U$ sends $I_{i_j}$ and $N_{i_j}$ to $B$. $B$ is then able to compute $\prod g(I_{i_j} \| N_{i_j})$ and compare it with the stored list of values produced from creating the coins. He can then determine which user corresponds to this value.

### 2.3.3 Creating a Coin

The Customer U wants to receive an electronic bill $C$ worth \$100 from the Bank $B$. The value of the bill corresponds to $(e'_B, n'_B)$. The following protocol is conducted to create the coin:

1. $U$ chooses a random number $b$. He then sends the following $Z$ to the bank:

$$Z = r^{e'_B} g(L \| b) \mod n'_B,$$

where $r \in_R \mathbf{Z}_{n'_B}$, where $\in_R$ means a random choice.

2. $B$ gives $Z^{d'_B} \mod n'_B$ to $U$ and charges \$100 to U's account.

3. $U$ extracts the electronic bill

$$C = (g(L\|b))^{d'_B} \mod n'_B.$$

### 2.3.4 Spending the Coin

To illustrate the spending protocol easily, we will follow an example from [9] of the Customer paying \$75 to the Merchant $M$ out of the \$100 he had received from the Bank. The protocol is based on a binary tree of three levels, as shown in Figure 2.1.

1. The Customer computes

$$\Gamma_{i_j,0} = \langle f_\Gamma(C\|0\|N_{i_j})\rangle_{QR}$$

   for $1 \leq j \leq K/2$.

2. $U$ computes

$$X_{i_j,00} = [\Gamma_{i_j,0}^{1/4} \mod N_{i_j}]_{-1} \quad \text{(corresponding to \$50)},$$

   and

$$X_{i_j,010} = [(\Omega_{i_j,0}^2 \Gamma_{i_j,0})^{1/8} \mod N_{i_j}]_{-1} \quad \text{(corresponding to \$25)},$$

   where $\Omega_{i_j,0} = \langle f_\Omega(C\|0\|N_{i_j})\rangle_1$, for $1 \leq j \leq K/2$.

3. $U$ sends $(I_{i_j}, N_{i_j}, X_{i_j,00}, X_{i_j,010})$ for $1 \leq j \leq K/2$, and $(L, C)$ to the Merchant.

   Figure 2.3 depicts the procedure.

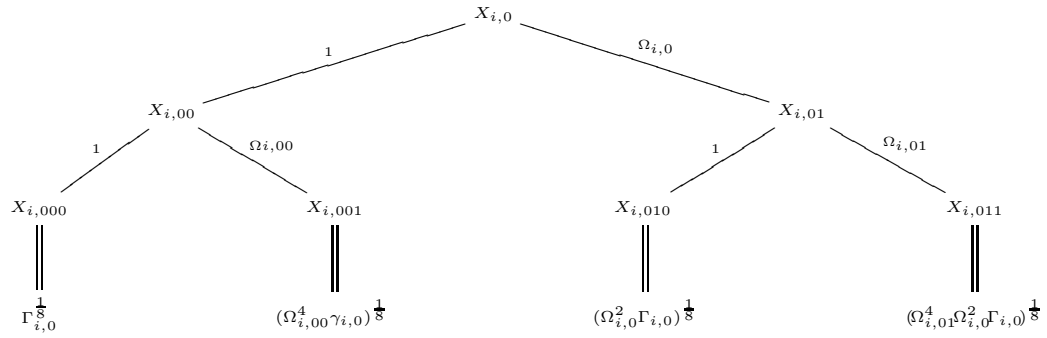


Figure 2.3: The $X$ values

Here are the definitions in the general case:

$$\Omega_{i,j_1\cdots j_t} = \langle f_\Omega(C\|j_1\|\cdots\|j_t\|N_i)\rangle_1$$
$$X_{i,j_1\cdots j_t} = \left[(\Omega_{i,j_1\cdots j_{t-1}}^{2^{t-1}j_t}\Omega_{i,j_1\cdots j_{t-2}}^{2^{t-2}j_{t-1}}\cdots\Omega_{i,j_1}^{2j_2}\Gamma_{i,0})^{1/2^t}\right]_{-1}.$$

The computation of the $2^t$th root is done by taking successive square roots mod $N_i$. At each stage except the last, the square root is chosen so as to be the one that is a square mod $N_i$.

4. $M$ verifies the validity of the license $L$ for $(I_{i_j}, N_{i_j})$, and the signature $C$ for $L$. The Merchant computes $\Omega_{i_j,0}$ and $f_\gamma(C\|0\|N_{i_j})$ for $1 \le j \le K/2$. He also verifies the validity of $X_{i_j,00}$, and $X_{i_j,010}$ by checking that the following statements hold:

$$(X_{i_j,00}/N_{i_j}) = (X_{i_j,010}/N_{i_j}) = -1$$
$$X_{i_j,00}^4 = d_{i_j} f_\Gamma(C\|0\|N_{i_j}) \mod N_{i_j}$$
$$X_{i_j,010}^8 = d_{i_j}\Omega_{i_j,0}^2 f_\Gamma(C\|0\|N_{i_j}) \mod N_{i_j},$$

where $d_{i_j} \in \{\pm 1, \pm 2\}$.

5. The Merchant randomly selects bits $E_{i_j,00}, E_{i_j,010} \in \{0,1\}$ for $1 \leq j \leq K/2$. He sends them to $U$.

6. The Customer computes

$$\Lambda_{i_j,00} = \langle f_\Lambda(C\|00\|N_{i_j})\rangle_{QR}$$

$$\Lambda_{i_j,010} = \langle f_\Lambda(C\|010\|N_{i_j})\rangle_{QR}$$

$$Y_{i_j,00} = [\Lambda_{i_j,00}^{1/2}]_{(-1)^{E_{i_j,00}}}$$

$$Y_{i_j,010} = [\Lambda_{i_j,010}^{1/2}]_{(-1)^{E_{i_j,010}}}$$

and sends $Y_{i_j,00}, Y_{i_j,010}$ to $M$, for $1 \leq j \leq K/2$.

7. $M$ verifies that

$$(Y_{i_j,00}/N_{i_j}) = (-1)^{E_{i_j,00}}$$

$$(Y_{i_j,010}/N_{i_j}) = (-1)^{E_{i_j,010}}$$

$$Y_{i_j,00}^2 = d'_{i_j} f_\Lambda(C\|00\|N_{i_j}) \mod N_{i_j}$$

$$Y_{i_j,00}^2 = d'_{i_j} f_\Lambda(C\|00\|N_{i_j}) \mod N_{i_j},$$

where $d'_{i_j}$ and $d''_{i_j}$ are chosen in $\{\pm 1, \pm 2\}$ so that the right-hand sides are squares mod $N_{i_j}$. If these equations are valid, $M$ accepts the \$75 coin from the Customer.

This protocol, of course, can be generalized to a case of withdrawing and spending any amount of money (up to the value of the bill $C$).

### 2.3.5 The Bank Credits the Merchant's Account

In order for the Bank to credit the Merchant's account by the requested amount, $M$ first sends to $B$ the history of the Customer's spending protocol shown earlier.

$B$ checks the validity of the history and then stores it in its database. As we'll see later, if $B$ sees a double payment, $B$ reveals the secret information, $S_{i_j}$ for some $j$, of the Customer. This contains $ID_U$, so the user is revealed.

## 2.4 Correctness

The protocol presented in [9] satisfies five of the criteria mentioned earlier, namely, (1) independence, (2) security, (3) privacy, (4) off-line payment, and (6) divisibility. It is quite obvious that the protocol satisfies (1) and (4).

(3) privacy: Since factoring is difficult for $B$ and $M$, they cannot obtain any knowledge about the identity of $U$ with non-trivial probability.

(6) divisibility: If the previously mentioned restrictions on the usage of the binary tree are satisfied, we have the divisibility condition satisfied.

**(2) security:** There are two possible types of double spending. First, the Customer can attempt to spend the same node in the binary tree twice. Second, he can attempt to spend two distinct nodes, where one node is an ancestor to the other.

**Case I: Two distinct nodes.** In this case, one node is a descendant of the other. We have two subcases. We will illustrate each by an example. For each node spent, a random set of indices $i_j$ was chosen. With high probability there will be overlap between the sets chosen for each spending. Let $i$ be an index in the overlap.

**(a)** The Customer spends nodes $n_{00}$ and $n_{000}$. The Customer gave

$$X_{i,00} = [\Gamma_{i,0}^{1/4}]_{-1}$$

$$X_{i,000} = [\Gamma_{i,0}^{1/8}]_{-1}$$

to the Merchants, who gave these to the Bank. Note that

$$X_{i,000}^2 = [\Gamma_{i,0}^{1/4}]_{QR}.$$

since the left side is a square. Then we observe that $X_{i,00}$ and $X_{i,000}^2$ are square roots of $\Gamma_{i,0}^{1/2}$. They cannot differ by sign since their Jacobi symbols are different. Therefore, the Bank can factor $N_i$. This allows the Bank to identify $U$, as follows.

The Customer U has sent $I_i$ to the Merchant. Note that $I_i = I_{1,i} \| I_{2,i}$. So the Bank receives $I_{1,i}$ and $I_{2,i}$. Also, $I_{1,i} \equiv S_{1,i}^2 \mod N_i$. $B$ has found the factorization of $N_i$, due to double-spending. $B$ can find the square root of $I_{1,i}$ mod $p_i$ and mod $q_i$. By the Chinese Remainder theorem, there are four choices for the square root of $I_{1,i}$ mod $N_i$. Hence there are four choices for $S_{1,i}$ and, similarly, four choices for $S_{2,i}$. So there are a total of 16 choices for $S_i = S_{1,i} \| S_{2,i}$. Since $S_i$ contains $ID_U$, there are at most sixteen choices for $ID_U$ (sixteen can be reduced because of the structure of $S_i$), and probably only one that corresponds to a person. Therefore, $U$ is caught.

**(b)** The Customer spends nodes $n_{00}$ and $n_{001}$. The Customer gave

$$X_{i,00} = [\Gamma_{i,0}^{1/4}]_{-1}$$

$$X_{i,001} = [(\Omega_{i,00}^4 \Gamma_{i,0})^{1/8}]_{-1}$$

to the Merchants, who gave these to the Bank. Note that

$$X_{i,001}^2 = [\Omega_{i,00} \Gamma_{i,0}^{1/4}]_{QR}.$$

since the left side is a square.

Then we observe that $X_{i,00}$ and $X_{i,000}^2/\Omega_{i,00}$ are square roots of $\Gamma_{i,0}^{1/2}$. Note that

$$\left((X_{i,000}^2/\Omega_{i,00})/N_i\right) = \left(\Omega_{i,00}/N_i\right) = 1$$

by the choice of $\Omega_{i,00}$. Therefore, the two square roots cannot differ by sign since their Jacobi symbols are different. Therefore, the Bank can factor $N_i$. This allows the Bank to identify $U$, as before.

**Case II: Spending one node twice.** For concreteness, call the node $n_{00}$. We will illustrate by an example. For each time the node is spent, a random set of indices $i_j$ was chosen. With high probability there will be overlap between the sets chosen for each spending. For each index $i$ in the overlap, each Merchant chose a challenge bit $E_i$. With reasonable probability, one of the indices in the overlap has different challenges, from different Merchants. Let $i$ be such an index. Say $E'_i = 0$ and $E'_i = 1$. Then, during the spending protocol, the following responses were sent to the Merchants, who gave them to the Bank:

$$Y'_{i,00} = [\Lambda_{i,00}^{1/2}]_1$$
$$Y''_{i,00} = [\Lambda_{i,00}^{1/2}]_{-1}.$$

The Bank then obtains two distinct square roots of $\Lambda_{i,00}$ and can factor $N_i$. This allows the Bank to identify $U$, as before.

## 2.5 Transferring Cash

Finally, Okamoto and Ohta demonstrate their system satisfying the criterion (5), transferability. We keep the earlier example used, where the value of the coin is \$100, customer $U_1$, who has spent \$75 of it, transfers the remaining \$25 to customer $U_2$ and $U_2$ uses \$25 at shop $M$. The protocol is identical except for one additional step, in which $U_1$ transfers $C$ to $U_2$. We describe this step in the following:

1. $U_2$ assumes the role of the Merchant in the earlier protocol. $U_1$ gives node $n_{011}$ to $U_2$.

2. $U_1$ digitally signs a message that includes the coin, the node it corresponds to, and the license of $U_2$. This information records the transfer in step 1 and protects both Customers.

3. The rest of the protocol is the same as the earlier one without the transfer, except that now $U_2$, not $U_1$, pays the Merchant \$25.

# Chapter 3

# Divisible Electronic Coins: Eng–Okamoto

## 3.1 Introduction

The paper [7] presents an improvement to the divisible off-line electronic cash scheme constructed by Okamoto and Ohta earlier. It is based on discrete logarithms rather than factorization.

Okamoto and Ohta's earlier electronic cash system has two sets of weakness: the amount of required communication between the bank and the merchant, and the needed memory size of the bank's database. Eng and Okamoto reduce the memory requirement to less than 1/10 of the Okamoto-Ohta system.

## 3.2 Preliminaries

Let $p$ and $q$ be large primes where $q \mid (p-1)$. Let $x \in_R X$ indicate that $x$ is randomly and uniformly selected from $X$. Concatenation is denoted by $\|$. If $b$ is a bit, $\overline{b}$ is the negation of $b$. Let $H : \{0,1\}^* \rightarrow \{0,1\}^{2|q|}$ be a polynomial-time

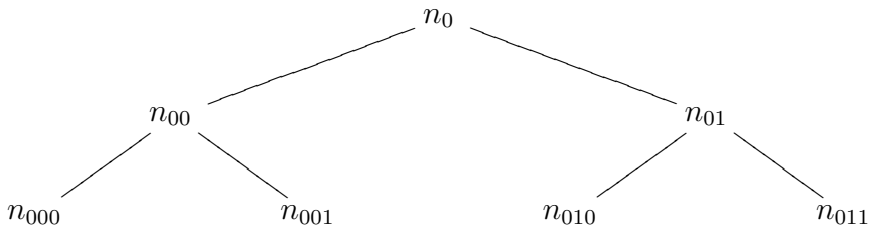


Figure 3.1: The nodes

computable one-way hash function.

Eng and Okamoto use a binary tree approach as Okamoto and Ohta did. The root node of a binary tree is denoted by $n_0$, and the remaining nodes are represented with subscripts 0s and 1s, where "0" represents a left branch and "1" represents a right branch. For example, the root node's children are written as $n_{00}$ and $n_{01}$. And their children are expressed as $n_{000}$ and $n_{001}$, and $n_{010}$ and $n_{011}$, respectively.

Similar to what Okamoto and Ohta did, each coin of worth $w$ is represented by a tree of $(1 + log_2 w)$ levels and $w$ leaves. The leaves are nodes that are at the bottom level of the binary tree, and the root node at the top represents the value $w$ of the coin. As one goes down each level, each node at the next level represents half of the value of its parent's node. This construction is repeated until the bottom row consists of nodes of worth \$1. For example, if a coin is worth \$8, then the tree has four levels and 8 bottom-nodes of \$1 each.

We will show later that we can extend this binary structure to dividing a coin into any number of pieces. For example, we could imitate the decimal structure of ordinary cash systems. However, this would be much harder to do in Okamoto and Ohta's system.

Divisibility can now be implemented under one rule: Once a node is spent, neither its parent nor any of its children nodes is available to be spent. This rule prevents double spending. If double spending occurs, the perpetrator is identified by the route that defines the node being double-spent.

## 3.3 The Electronic Cash Scheme

### 3.3.1 Initialization

The initialization is based on the Okamoto–Ohta scheme.

1. The Customer $U$ has a public key $m = g_1^{x_1} g_2^{x_2} \mod p$.

2. $U$ proves to the bank that it knows the private key $(x_1, x_2)$ using the following protocol:

   (a) The customer randomly selects $(r_1, r_2)$ and calculates $\beta = g_1^{r_1} g_2^{r_2} \mod p$.

   (b) The Bank $B$ returns a challenge message $\alpha$, which is different each time. Note that the bank has no information about $(x_1, x_2)$.

   (c) The Customer sends $y_1 = r_1 + \alpha x_1 \mod q$ and $y_2 = r_2 + \alpha x_2 \mod q$ to the bank who verifies that

$$g_1^{y_1} g_2^{y_2} \equiv \beta m^{\alpha} \mod p.$$

### 3.3.2 Creating a Coin

A spender wants to withdraw $w = 2^{\ell}$ from his account. The protocols are based on an earlier scheme of Brands.

Let $p, q, g, g_1, g_2$ be system parameters provided by the Bank $B$, where $g$, $g_1$, and $g_2$ have order $q$ mod $p$. Let $I = g_1^u \mod p$ be the identity of the Customer $U$, where $u$ is $U$'s secret key. Let $h = g^x \mod p$ be the public key of the Bank, where $x$ is $B$'s secret key.

The protocol proceeds as follows:

1. $U$ conducts the initialization stage and obtains $T = g_1^{r_{0,1}} g_2^{r_{0,2}} \mod p$. The Customer $U$ sends his identity, $I = g_1^u \mod p$, to Bank $B$. Here, $u$ is the secret identity number of $U$, and $I$ is his public number. If someone finds $u$, they can identify $U$.

2. $B$ subtracts $w = 2^L$ dollars from $U$'s account. $B$ chooses $w \in_R \mathbf{Z}_q$ and sends $z = m^x \mod p$, $a = g^w \mod p$, $b = m^w \mod p$ to $U$.

3. $U$ randomly selects $s, t, v \in_R \mathbf{Z}_q$ and calculates $m' = m^s \mod p$, $z' = z^s \mod p$, $a' = a^t g^v \mod p$, $b' = b^{st}(m')^v \mod p$, $c' = H(m', z', a', b', T)$, and $c = c'/t \mod q$. Then $U$ sends $c$ to $B$. Note that $s$ is used to mask $m$.

4. $B$ sends $r = xc + w \mod q$ to $U$.

5. $U$ verifies the validity of $z, a, b$ by checking that $m^r = z^c b \mod p$ and $g^r = h^c a \mod p$, and then calculates $r' = rt + v \pmod q$.

### 3.3.3 Computation of $t$-values and $r$-values for Leaf Nodes

The Customer chooses a random value $e$ as a secret seed value. Every node is assigned a $t$-value. Let $n_{0j_1j_2\cdots j_v}$, where $j_i \in \{0, 1\}$, be a node. Its $t$-value is denoted by $t_{0j_1j_2\cdots j_v}$. If $n_{0j_1j_2\cdots j_\ell}$ is a leaf of the tree, its $t$-value is defined to be

$$t_{0j_1j_2\cdots j_\ell} = \mathcal{H}(e \| 0j_1j_2 \cdots j_\ell).$$

If we have defined the $t$-values of a node, we can define two "$r$-values" of the node by

$$t_{0j_1j_2\cdots j_v} = (r_{0j_1j_2\cdots j_v,1} \| r_{0j_1j_2\cdots j_v,2}) \quad \text{where} \quad r_{0j_1j_2\cdots j_v,i} \in \{0,1\}^{|q|}.$$

The $t$-values of non-leaf nodes are determined inductively from the $r$-values of their children. If the $t$-values of a non-leaf node's left and right children are

$$t_{0j_1j_2\cdots j_v0} = (r_{0j_1j_2\cdots j_v0,1} \| r_{0j_1j_2\cdots j_v0,2})$$

$$t_{0j_1j_2\cdots j_v1} = (r_{0j_1j_2\cdots j_v1,1} \| r_{0j_1j_2\cdots j_v1,2})$$

then its $t$-value is defined to be

$$t_{0j_1j_2\cdots j_v} = \mathcal{H}(\mathcal{H}(g_1^{r_{0j_1j_2\cdots j_v0,1}} g_2^{r_{0j_1j_2\cdots j_v0,2}}) \| \mathcal{H}(g_1^{r_{0j_1j_2\cdots j_v1,1}} g_2^{r_{0j_1j_2\cdots j_v1,2}})).$$

In this fashion, we obtain $t$-values for all nodes, all the way up to the root node.

### 3.3.4 Spending the Coin

Spending the coin requires two stages in Eng and Okamoto's scheme. The first stage is coin authentication.

1. The Customer $U$ gives $m' = g_1^{us} g_2^{s} \mod p$, $T$ (computed when the coin was created), and the signature $sign(m', T) = \{z', a', b', r'\}$ to the Merchant.

2. $M$ checks that $m' \neq 1 \mod p$.

3. $M$ computes $c'$ and verifies that the following three equations hold:

$$g^{r'} \equiv h^{c'} a' \mod p$$

$$m^{r'} \equiv (z')^{c'} b' \mod p$$

$$c' \equiv \mathcal{H}(m', z', a', b', T)$$

This completes the Merchant's verification of the Bank's signature on $m'$ and $T$.

The next stage is what Eng-Okamoto call "denomination revelation," where the Customer $U$ reveals information on the nodes of the tree that represent the coin. Suppose he spends the node $n_{0j_1j_2\cdots j_k}$.

1. $U$ reveals

$$\beta_{0j_1j_2\cdots j_k} = g_1^{r_{0j_1j_2\cdots j_k,1}} g_2^{r_{0j_1j_2\cdots j_k,2}} \mod p,$$

   which is the node's contribution to the $t$-value of its ancestor node.

2. $U$ then reveals the information that is used to compute the $t$-values for all ancestor nodes of $n_{0j_1j_2.j_k}$. Namely,

$$\mathcal{H}(g_1^{r_{0j_1j_2\cdots\overline{j_k},1}} g_2^{r_{0j_1j_2\cdots\overline{j_k},2}})$$
$$\mathcal{H}(g_1^{r_{0j_1j_2\cdots\overline{j_{k-1}},1}} g_2^{r_{0j_1j_2\cdots\overline{j_{k-1}},2}})$$
$$.$$
$$.$$
$$.$$
$$\mathcal{H}(g_1^{r_{0j_1\overline{j_2},1}} g_2^{r_{0j_1\overline{j_2},2}})$$
$$\mathcal{H}(g_1^{r_{0\overline{j_1},1}} g_2^{r_{0\overline{j_1},2}}).$$

3. Then $M$ travels up the tree from the node $n_{0j_1j_2\cdots j_k}$. He is able to compute all its ancestors' $t$-values including the root $t$-value $t_0$. Then he can verify $T$ and the signature of $m'$.

4. $M$ presents a challenge $\alpha \in \mathbf{Z}_p^*$, which is a function of the date, time, Merchant's identity, and other variants.

5. $U$ responds with

$$y_1 = r_{0j_1j_2\ldots j_k,1} + \alpha us \mod q$$

$$y_2 = r_{0j_1j_2\ldots j_k,2} + \alpha s \mod q$$

6. $M$ verifies that

$$(g_1)^{y_1}(g_2)^{y_2} \equiv \beta_{0j_1j_2\ldots j_k}(m')^{\alpha} \mod p.$$

This technique is quite similar to a zero knowledge proof, where the Customer is able to prove that he has a legitimate coin without revealing too much information. Only the $t$-values and $r$-values of the ancestor nodes of $n_{0j_1j_2.j_k}$ are revealed to $M$ and $B$.

Suppose the coin $(m', T, sign(m', T))$ is worth $2^{\ell}$ and that $U$ wants to spend $\$x$ from it. We suppose that the binary representation of $x$ is $b_1b_2\ldots b_{\ell+1}$ in the $\ell+1$ level binary tree.

Let $\nu = \#\{b_i \mid b_i = 1, 1 \leq i \leq \ell+1\}$ be the Hamming weight of $x$. It indicates the number of nodes that will be used for spending $\$x$. Whenever $b_i = 1$, it says that a node at the $i$-th level is used. For example, if $x = 0100100$, then $\ell = 6$ and $\nu = 2$. So, he can use a node at the 2nd level, say, $n_{00}$, and a node at the 5th level, say, $n_{01000}$.

$U$ sends the following to the Merchant.

$$\beta_{00} = g_1^{r_{00,1}} g_2^{r_{00,2}} \mod p$$

$$\beta_{01000} = g_1^{r_{01000,1}} g_2^{r_{01000,2}} \mod p$$

$$\mathcal{H}(g_1^{r_{01001,1}} g_2^{r_{01001,2}} \mod p)$$

$$\mathcal{H}(g_1^{r_{0101,1}} g_2^{r_{0101,2}} \mod p)$$

$$\mathcal{H}(g_1^{r_{011,1}} g_2^{r_{011,2}} \mod p)$$

Observe that there will be $\nu$ values of $\beta$ from Step 1 of this protocol. In the example of $x = 0100100$, there will be two $\beta$ values. If $\ell'$ is the smallest value for which $b_i = 0$ for all $i > \ell'$, and if the $\nu$ nodes that are used are optimally selected, then $(\ell' - \nu)$ hashed values are revealed. In the example of $x = 0100100$, three $(= 5 - 2)$ hash values are revealed.

### 3.3.5 The Bank Credits the Merchant's Account

Crediting the Merchant's account is done the same way as in Okamoto and Ohta's scheme, by sending the transcript of the transaction to the Bank.

## 3.4 Security

The goal of the security is to ensure that the Customer's identity is protected, except when a coin is incorrectly spent. There are two possible types of double spending. First, the Customer can attempt to spend the same node in the binary tree twice. Second, he can attempt to spend two distinct nodes, where one node is an ancestor to the other. We use an example of spending \$1 out of a \$4-coin to illustrate how both types of double spending are caught by the Eng-Okamoto scheme.

Since the coin is worth \$4, there are four leaf-nodes and their $t$-values are computed in the following way:

$$t_{000} = \mathcal{H}(e\|000), \quad t_{001} = \mathcal{H}(e\|001), \quad t_{010} = \mathcal{H}(e\|010), \quad t_{011} = \mathcal{H}(e\|011).$$

for a randomly selected value $e$. As shown earlier, the $t$-values of the remaining

nodes are computed as follows:

$$\begin{aligned}
t_{0j_1j_2} &= (r_{0j_1j_2,1} \| r_{0j_1j_2,2}) \quad \text{for all } j_1, j_2 \in \{0,1\} \\
t_{01} &= \mathcal{H}(\mathcal{H}(g_1^{r_{010,1}} g_2^{r_{010,2}}) \| \mathcal{H}(g_1^{r_{011,1}} g_2^{r_{011,2}})) \\
&= (r_{01,1} \| r_{01,2}) \\
t_{00} &= \mathcal{H}(\mathcal{H}(g_1^{r_{000,1}} g_2^{r_{000,2}}) \| \mathcal{H}(g_1^{r_{011,1}} g_2^{r_{011,2}})) \\
&= (r_{01,1} \| r_{01,2}) \\
t_0 &= \mathcal{H}(\mathcal{H}(g_1^{r_{00,1}} g_2^{r_{00,2}}) \| \mathcal{H}(g_1^{r_{01,1}} g_2^{r_{01,2}}))
\end{aligned}$$

We can now proceed to examine two possible scenarios of double spending.

**Case 1. Spending the same node twice.** Suppose $U$ tries to spend $n_{001}$ twice. During the spending protocol, we saw that $n_{001}$'s contribution to the $t$-value of its ancestor node, $\beta_{001}$, is revealed. We also note that hash values of its sibling and of the siblings of all its ancestor nodes are also revealed, namely, $\mathcal{H}(g_1^{r_{000,1}} g_2^{r_{000,2}})$ and $\mathcal{H}(g_1^{r_{01,1}} g_2^{r_{01,2}})$. When the Merchant presents a challenge, $\alpha$, in the first spending, and another challenge, $\alpha'$, in the second spending, the Customer returns two sets of responses. In the first spending, he returns $y_1 = r_{001,1} + \alpha us \mod q$ and $y_2 = r_{001,2} + \alpha s$. In the second spending, he returns $y_1' = r_{001,1} + \alpha' us \mod q$ and $y_2' = r_{001,2} + \alpha' s$. Then by solving the system of equations, the Merchant and the Bank can retrieve $u$ and $s$.

**Case 2. Spending a node's child node.** Suppose $U$ spends $n_{00}$ and then tries to spend its child node, $n_{001}$. From the spending of $n_{00}$, the Customer returns the response to a challenge. Namely, $y_1 = r_{00,1} + \alpha us$ and $y_2 = r_{00,2} + \alpha s$. If $U$ then spends $n_{001}$, the Customer reveals $\beta_{001}$ and $\mathcal{H}(\beta_{000})$. Note that

$$\mathcal{H}(\mathcal{H}(\beta_{000}) \| \mathcal{H}(\beta_{001})) = t_{00} = (r_{00,1} \| r_{00,2}).$$

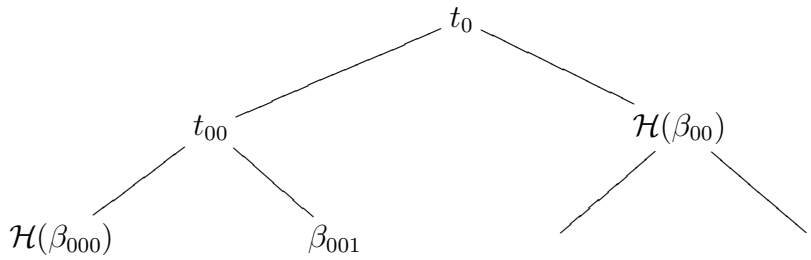


Figure 3.2: Node $n_{001}$ is spent

This enables the Merchant to deduce the values of $u$ and $s$, since he now knows $y_1, y_2, r_{00,1}$ and $r_{00,2}$. So the violation is caught.

Figure 3.2 shows what is revealed when node $n_{001}$ is spent.

## 3.5 Higher Divisibility

We now extend the Eng-Okamoto scheme to a more general case. Suppose we divide each node into three children. We will demonstrate the analogous two cases of double spending. Observe that the $t$-values of non-leaf nodes are now computed by three beta-values of their children, while each node still has only two r-values.

More generally, a similar technique can be applied to dividing a coin into any number of pieces, where the number of pieces does not have to be the same at each node. This allows us to mimic the real cash system of U.S. dollars, quarters, dimes, nickels, and pennies.

The Customer chooses a random value $e$ as a secret seed value. Every node is assigned a $t$-value. Let $n_{0j_1j_2\cdots j_v}$, where $j_i \in \{0, 1, 2\}$, be a node. Its $t$-value is

| | $U$ | $M$ | $B$ |
|---|---|---|---|
| Creating Coin | $u, T, m, z, a,$ $b, s, t, v, m',$ $z', a', b'c, c',$ $t$-values, $r$-values, $\beta$-values | | $I, m, w, z,$ $a, b, c, r$ |
| Spending Coin | $\alpha, y_1, y_2$ | $m', T,$ $sign(m', T),$ $t$-values above node, $r$-values above node, $\beta$ of the node, $\alpha, y_1, y_2$ | |
| Depositing Coin | | | $m', T,$ $sign(m', T),$ $t$-values above node, $r$-values above node, $\beta$ of the node, $\alpha, y_1, y_2$ |

Table 3.1: Who knows what during the Eng-Okamoto protocol

denoted by $t_{0j_1j_2\cdots j_v}$. If $n_{0j_1j_2\cdots j_\ell}$ is a leaf of the tree, its $t$-value is defined to be

$$t_{0j_1j_2\cdots j_\ell} = \mathcal{H}(e\|0j_1j_2\cdots j_\ell).$$

If we have defined the $t$-values of a node, we can define two "$r$-values" of the node by

$$t_{0j_1j_2\cdots j_v} = (r_{0j_1j_2\cdots j_v,1}\|r_{0j_1j_2\cdots j_v,2}) \quad \text{where} \quad r_{0j_1j_2\cdots j_v,i} \in \{0,1\}^{|q|}.$$

The $t$-values of non-leaf nodes are determined inductively from the $r$-values of their children. If the $t$-values of a non-leaf node's left, middle, and right children are

$$t_{0j_1j_2\cdots j_v0} = (r_{0j_1j_2\cdots j_v0,1}\|r_{0j_1j_2\cdots j_v0,2})$$

$$t_{0j_1j_2\cdots j_v1} = (r_{0j_1j_2\cdots j_v1,1}\|r_{0j_1j_2\cdots j_v1,2})$$

$$t_{0j_1j_2\cdots j_v2} = (r_{0j_1j_2\cdots j_v2,1}\|r_{0j_1j_2\cdots j_v2,2})$$

then its $t$-value is defined to be

$$t_{0j_1j_2\cdots j_v} =$$

$$\mathcal{H}(\mathcal{H}(g_1^{r_{0j_1j_2\cdots j_v0,1}} g_2^{r_{0j_1j_2\cdots j_v0,2}})\|\mathcal{H}(g_1^{r_{0j_1j_2\cdots j_v1,1}} g_2^{r_{0j_1j_2\cdots j_v1,2}})\|\mathcal{H}(g_1^{r_{0j_1j_2\cdots j_v2,1}} g_2^{r_{0j_1j_2\cdots j_v2,2}})).$$

In this fashion, we obtain $t$-values for all nodes, all the way up to the root node.

The spending protocol is identical to the binary structure, except that the hash values of both its siblings and of both of the siblings of each of its ancestor nodes are also revealed.

We can now proceed to examine two possible scenarios of double spending.

**Case 1. Spending the same node twice.** Suppose $U$ tries to spend $n_{000}$ twice. During the spending protocol, we saw that $n_{000}$'s contribution to the $t$-value of its ancestor node, $\beta_{000}$, is revealed. We also note that hash values of both

of its siblings and of both of the siblings of each of its ancestors are also revealed, namely, $\mathcal{H}(g_1^{r_{001,1}} g_2^{r_{001,2}})$, $\mathcal{H}(g_1^{r_{002,1}} g_2^{r_{002,2}})$, $\mathcal{H}(g_1^{r_{01,1}} g_2^{r_{01,2}})$, and $\mathcal{H}(g_1^{r_{02,1}} g_2^{r_{02,2}})$. When the Merchant presents a challenge, $\alpha$, in the first spending, and another challenge, $\alpha'$, in the second spending, the Customer returns two sets of responses. In the first spending, he returns $y_1 = r_{000,1} + \alpha us \mod q$ and $y_2 = r_{000,2} + \alpha s$. In the second spending, he returns $y_1' = r_{000,1} + \alpha' us \mod q$ and $y_2' = r_{000,2} + \alpha' s$. Then by solving the system of equations, the Merchant and the Bank can retrieve $u$ and $s$.

**Case 2. Spending a node's child node.** Suppose $U$ spends $n_{00}$ and then tries to spend its child node, $n_{000}$. From the spending of $n_{00}$, the Customer returns the response to a challenge. Namely, $y_1 = r_{00,1} + \alpha us$ and $y_2 = r_{00,2} + \alpha s$. If $U$ then spends $n_{000}$, the Customer reveals $\beta_{000}$, $\mathcal{H}(\beta_{001})$, and $\mathcal{H}(\beta_{002})$. Note that

$$\mathcal{H}(\mathcal{H}(\beta_{000}) \| \mathcal{H}(\beta_{001}) \| \mathcal{H}(\beta_{002})) = t_{00} = (r_{00,1} \| r_{00,2}).$$

This enables the Merchant to deduce the values of $u$ and $s$, since he now knows $y_1, y_2, r_{00,1}$ and $r_{00,2}$. So the violation is caught.

# Chapter 4

# Some Other Systems

## 4.1 Introduction

We will now look at a few more systems where they claim to provide untraceability or unlinkability. We will then study some papers that found flaws in these systems.

## 4.2 Damgård's Scheme

In Crypto '90, Damgård [5] presented an untraceable online payment system with provable security against abuse by individuals. Pfitzmann and Waidner [11] show how to break the untraceability of Damgård's payment system. They also introduce possibilities to improve it further with new features.

In untraceable payment systems, the Bank is needed to approve each payment. However they do not allow the Bank to observe the activities of the individuals. For example, when individuals withdraw money from one account and deposit it into another account, the Bank should not be able to trace the activity to find out the identities of the sender and the receiver. The security of the systems

relies, for example, on an RSA scheme so that one cannot double spend or create coins themselves. Pfitzmann and Waidner show that Damgård's payment system is not untraceable at all.

Damgård's system uses a provably secure signature scheme. The Customer gives $m'$ and the signature $(m', T, sign(m', T))$ to the Merchant, who gives it to $B$. During the depositing protocol, the Bank is able to obtain these numbers, $m'$ and $(m', T, sign(m', T))$. Then $B$ can store the history and would be able to trace the identity of the Customer.

Essentially, the system enables the Bank to see the signature. This gives the Bank ways of tracing the coin in some cases.

Pfitzmann and Waidner repair the untraceability flaw of Damgård's system by requiring the Customer to use a zero-knowledge proof to show that he has the signature during the deposit protocol.

## 4.3 D'Amiano and Di Crescenzo's Cash System

D'Amiano and Di Crescenzo [6] claim to present an electronic cash system that provide complete untraceability. They also claim that the cash maintains its size as it gets transferred from one person to another. Pfitzmann, Schunter, and Waidner [10] break this system and show that D'Amiano and Di Crescenzo's scheme does not provide any untraceability. They identify how untraceability is not secured in D'Amiano and DiCrescenzo's system without any growth in size of the coin.

### 4.3.1 The System

The initialization and the withdrawal protocols are almost identical to those of Okamoto-Ohta and Eng-Okamoto. When a double-spending occurs, the Bank receives the same coin twice in two different deposits. $B$ can then broadcast a request for all previous users of the coin to send in the transcripts or information they have about the coin. Then the Bank is able to determine all previous owners of the coin. Either

1. one of the users who does not cooperate is detected,
2. one of the users who paid the coin twice is detected, or
3. two withdrawals of the same coin are detected.

In case (1), the attacker who does not cooperate is presumed to be guilty. In case (2), the Bank has detected what it had hoped to find, whereas case (3) does not pose any problem since the money has been withdrawn from the account.

### 4.3.2 The Untraceability Flaw

Pfitzmann and Waidner show that the untraceability is not ensured at all in this system. Suppose that there are two attackers in the system. In Figure 4.1, the attackers are denoted as $A_1$ and $A_2$, and the honest customers are $U_1$, $U_2$, $U_3$, and $U_4$. Each arrow indicates a payment. Suppose $A_2$ tries to double-spend, as indicated by two arrows originating from $A_2$. When the Bank traces the coin back, $A_2$ and $U_4$ both show they received the coin from $U_2$. The coins that came from $A_1$ are completely identical, so $U_2$ cannot prove that the coin was given to him twice. $A_1$ remains quiet without revealing that he got the coin from $A_2$, so

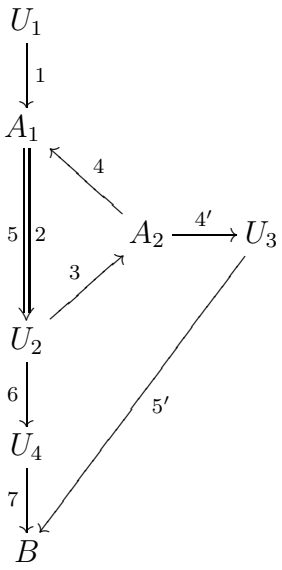


Figure 4.1: A double-spending scenario

$A_2$ is completely free of risk. So, $U_1$ is incorrectly detected to be double-spending and punished.

Note that if $A_2$ did not double-spend, $U_2$ would still have been given the same coin twice as shown in Figure 4.1. This would have been a normal transaction. Even if $U_2$ detects that a coin he once spent came back to him, he would not find it suspicious. In real paper cash system, the same can occur. Once the Customer spends a paper bill, it may come back to him at one point.

Pfitzmann and Waidner argue that a different signature must be given when paying a coin to the same person. Interactively, this can be done with a challenge or a transaction number. Transaction numbers and comparing a coin with all previously received ones are less efficient but are possible ways to prevent this type of attack. Non-interactively, time stamps can be used.

It is important to note that Chaum and Pedersen [4] make an even stronger

observation about the size of the cash. They show that it is impossible to ensure the transferability without growing the size of the coin. This shows that D'Amiano-Di Crescenzo's scheme is fundamentally flawed.

## 4.4 Canard and Gouget's Scheme

Canard and Gouget [2] present an off-line divisible e-cash scheme with unlinkability. They present security tags that can protect that anonymity of honest customers and that can reveal the identity only when the Customer cheats. This is done without using a trusted third party.

The main feature of their scheme is the use of non-interactive zero-knowledge proofs. They are used to verify the possession of various information without transferring any content information. The beauty of the non-interactive proofs is that they can be reproduced. The transcripts of the proofs that the Customer has the information are sent. This solves the linkability problem, commonly found in other systems. It is not obvious, however, how we can improve this system to make the cash transferable.

# Chapter 5

# Multiply Spendable Coins

The paper [8] by Ferguson presents the idea of $n$-spendable coins, where each coin can be spent up to $n$ times without revealing the Customer. $U$ is revealed, however, when the coin is used for the $n+1$st time.

The advantage of allowing $U$ to spend a coin several times by this scheme is its efficiency. It requires less storage than using multiple 1-spendable coins. Especially for systems like subway fare, the idea is useful, even though linkability is not prevented.

Ferguson generalizes the case of 1-spendable coins by using a secret sharing scheme. The generalization introduces the use of a higher degree polynomial to mask the identity of $U$.

$U$ has a polynomial $u + \sum_{i=1}^{n} k_i X^i$ mod a prime $q$, where $u$ is the secret identity number of $U$ and the secret numbers $k_1, \ldots, k_n$ are produced during the initialization stage.

Then the challenge-response consists of the following steps:

1. $M$ sends a challenge $x \in_R \mathbf{Z}_q$.

2. $U$ sends a value of the polynomial: $r = u + \sum_{i=1}^{n} k_i x^i$.

If the coin is spent $\ell$ times, $U$ reveals $\ell$ points on the graph of the polynomial. If $\ell \leq n$, this information does not reveal $u$. However, if $U$ spends the coin for the $n+1$st time, his identity is easily revealed by being able to construct the polynomial from the $n+1$ points.

Eng-Okamoto's case is a variation of the case when $n = 1$. $M$ sends a challenge $\alpha$ to $U$. Then $U$ sends the value at $X = \alpha$ of the linear polynomials $r_{000,1} + sX$ and $r_{000,2} + usX$. Suppose $U$ double-spends the coin. Then, with the second challenge $\alpha'$, the coefficients of the polynomial are revealed.

# BIBLIOGRAPHY


[1] Stefan Brands, "Untraceable off-line cash in wallets with observers," *Advances in Cryptology – CRYPTO '93*, Lecture Notes in Computer Science, vol. 576, Springer-Verlag, 1994, pp. 302-318.

[2] Sébastien Canard and Aline Gouget, "Divisible E-cash Systems can be Truly Anonymous," *Advances in Cryptology - EUROCRYPT 2007*, Springer Lecture Notes in Computer Science, vol. 4515, 2007, pp. 482-497.

[3] David Chaum, "Blind signatures for untraceable payments," *Advances in Cryptology - CRYPTO '82*, Plenum, New York, 1983, 199-203.

[4] David Chaum and Torben Pedersen, "Transferred Cash Grows in Size," *Advances in Cryptology – EUROCRYPT '92*, Lecture Notes in Computer Science, vol. 658, Springer-Verlag, 1993, pp. 89-105

[5] Ivan Damgård, "Payment Systems and Credential Mechanisms with Provable Security," *Advances in Cryptology – CRYPTO '88*, Lecture Notes in Computer Science, vol. 403, Springer-Verlag, 1990, pp. 328-335.

[6] Stefano D'Amiano and Giovanni Di Crescenzo, "Methodology for Digital Money based on General Cryptographic Tools," *Pre-proceeding of Eurocrypt '94*, 1994, pp. 151-162.

[7] Tony Eng and Tatsuki Okamoto, "Single-Term Divisible Electronic Coins," *Advances in Cryptology - EUROCRYPT '94*, vol. 950, Springer-Verlag, 1995, pp. 306-319.

[8] Niels Ferguson, "Extensions of Single-term Coins," *Advances in Cryptology - EUROCRYPT '93*, Lecture Notes in Computer Science, vol. 765, 1994, pp. 318-328.

[9] Tatsuki Okamoto and Kazuo Ohta, "Universal Electronic Cash," *Advances in Cryptology – CRYPTO '91*, Lecture Notes in Computer Science, vol. 576, Springer-Verlag, 1992, pp. 324-337.

[10] Birgit Pfitzmann, Matthias Schunter, and Michael Waidner, "How to Break and Repair another 'Provably Secure' Payment System," *Advances in Cryptology, EUROCRYPT '95* , Lecture Notes in Computer Science, vol. 921, Springer-Verlag, 1995, pp. 121-132.

[11] Birgit Pfitzmann and Michael Waidner, "How to Break and Repair a 'Provably Secure' Untraceable Payment System," *Advances in Cryptology, CRYPTO '91*, Lecture Notes in Computer Science, vol. 576, Springer-Verlag, 1992, pp. 338-350.

[12] Wade Trappe and Lawrence Washington, *Introduction to Cryptography with Coding Theory, 2nd edition*, Prentice Hall, 2006.

[13] "Electronic money" *Wikipedia, The Free Encyclopedia*, Wikimedia Foundation, Inc. 5 Aug 2008 <http://en.wikipedia.org/w/index.php?title=Electronic_money&oldid=228444788>.