ELECTRONIC MONEY AND THE DERIVED APPLICATIONS: ANONYMOUS MICROPAYMENT, RECEIPT-FREE ELECTRONIC VOTING AND ANONYMOUS INTERNET ACCESS

By

Chan Yuen Yan

A Thesis

submitted in partial fulfillment of the requirements

for the Degree of Master of Philosophy

Division of Information Engineering

The Chinese University of Hong Kong

June 2000

摘要

我們現正身處於數碼時代，很多日常應用已經由它們的傳統模式轉形至電子及在線模式。事實上，電子商貿的蓬勃發展將很多有利的改變帶入我們的日常生活中，此等改變包括增加了的選擇項目，效率以及方便。與此同時，我們需求一種新的貨幣以支持數碼經濟體系的流暢進展。

在這篇論文裏，我們對電子貨幣進行了一項詳盡的研究，這研究是從貨幣的歷史開始。當中所涉及到的密碼學(Cryptography)，例如公匙(public key)、電子證書(digital certificate)及先分後選(cut-and-choose)方法亦有介紹。電子現金(electronic cash)及微額付款(micropayment)協定的模型亦有仔細地被解釋。

我們在這論文中也有提出一種新的微額付款模式：不記名微額付款票(anonymous micropayment ticket)。可是，電子現金所作出的貢獻並不限於作爲一種數碼交易媒介。我們由電子現金協定中研究出兩種創新的密碼學應用：分別是不可轉用的電子投票通行証(non-transferable electronic voting pass)及不記名的上網服務(anonymous Internet access service)。這些應用皆會在這論文中闡述。

Abstract

As we are now situated in the Digital Age, many of the everyday applications have been transformed from their traditional formats into the electronic and on-line ones. In fact, the blooming of electronic commerce has brought many beneficial changes to our daily lives, which include increased number of choices, efficiency as well as convenience. At the same time, a new form of money is demanded so that it can support the smooth running of the digital economy.

In this thesis, we make a thorough study on electronic money, which begins from the history of money. The cryptography involving, such as public key digital signatures and cut-and-choose methodology are also introduced. Basic models of electronic cash and micropayment protocols are explained in details.

We also proposed a new format of micropayment: the Anonymous Micropayment Ticket. However, the contribution of electronic cash is not limited to being a digital medium of transaction. In this thesis, two more innovative cryptographic applications, namely the Non-Transferable Electronic Voting Passes and the Anonymous Internet Access Services are derived from the electronic cash protocols. All of these applications are described in this thesis.

Acknowledgement

I would like to thanks my supervisor Professor Victor K. W. Wei for his resourceful guidiance on my research process. This thesis would not have been possible without his valueable ideas and teaching. I would also like to thanks my family for giving me so much care and support.

Also need to be mentioned are the classmates who have been frighting with me together during these two years. Without their presence and support, school life would not have been so beautiful like this! They include Ivan who gives me the panadols when I feel headache; Joseph who drives me home when I work overtime; Hung Chai and Ah Yung who chat with me when I feel boring; Anson, Jimmy, Ah Tong and many of them who play Mid Town Madness with me so that I can relax from the stressful works. They all add colours onto my picture of university life.

Last but not least, may I say thank you to Jesus who always keeps me and loves me. Thank you Jesus, thank you Lord!

Contents

List of Tables

List of Figures

Chapter 1

Introduction

We are now living in the Digital Society. Many of the last decade researches have already turned into everyday applications. Examples include a numerous of electronic commerce applications and the WAP mobile phone. On 7 March 2000, the first official Internet voting was conducted in Arizona, which made a remarkable milestone for the Digital Age.

Suitable payment systems are necessary for supporting the transaction of goods and services in the Digital Economy. In fact the invention of electronic money greatly enhances the smooth running of electronic commerce. In the early days of human history when money is not yet invented, people practiced barter. Later physical commodities such as seashells, animals and tobacco were accepted as a medium of transaction, while the Lydians made the first coins in 700 BC. Since then the formats of money are being developed and modified continuously.

The first proposal of using numbers to represent coins was made by David Chaum in 1983 [1]. According to this paper, an electronic cash system consists of

electronic coins which are numbers. Today there are hundreds of proposed electronic cash systems, while some of them are circulating in the market. In fact, electronic payment systems are the most suitable form of transaction for electronic commerce because they can be built and run on the same media.

According to a survey conducted in June 1999 [2], the global electronic commerce market would reach 300 billion US dallars by the end of 2000, as shown in figure 1.1.

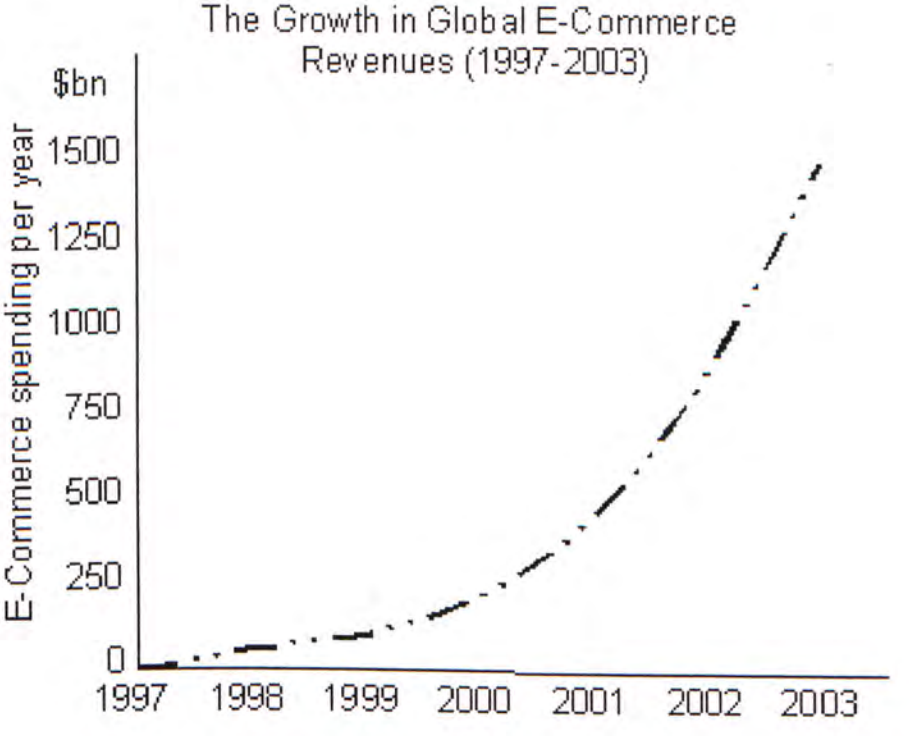

Figure 1.1: The Growth in Global E-Commerce Revenues (1997-2003)

Therefore the demand for more convenient and secure electronic cash systems is ever increasing.

1.1 Transition to a New Monetary System

Money is a widely accepted medium of exchange, which is essential for a smoothly-running market system. Along with the growth of the digital economy, electronic money is also gaining its significance. Government fiat money is likely to be replaced by stored value cards and electronic cash. An everyday example is the popularity of the Octopus system, which is a stored value card system designed for the public transportation system in Hong Kong. Another example is the Mondex [3] system which is now becoming popular among the university campuses.

1.2 Security and Cryptography

Security is an essential concern in the monetary system. In the paper-based monetary system, for example, special features are included so as to prevent counterfeiting. Extra cost on security guard is also required when transferring the money from one location to another.

In electronic money systems, a comparable level of security is also required, which is realized by cryptography. Therefore, electronic cash researches fall into the category of Cryptography because a lot of cryptographic primitives are involved in the design of an electronic cash systems. For example, the official watermark on paper cash is achieved by digital signatures electronically, while zero-knowledge proofs are employed in double spending prevention.

1.3 Electronic Cash: More than an Electronic Medium of Transaction

While electronic cash plays a very important role in the digital economy, its innovations are not limited to merely a digital medium of transaction. Different applications such as *Electronic Voting*, *Micropayment Systems* and *Anonymous Internet Access* are derived from the electronic cash protocols. These applications share some similar cryptographic techniques that have been employed in electronic cash protocols. In this thesis, a thorough study on electronic cash and the underlying cryptography is made. Besides, three other innovative applications are proposed, namely

1. The anonymous micropayment tickets
2. Non-transferable electronic voting passes
3. Anonymous Internet access services via proxy servers

These applications are motivated and derived by the electronic cash protocols.

In the first application, a new concept of electronic payment: micropayment in form of an electronic ticket is proposed. The micropayment ticket is deigned for transactions of small amount. In addition, unlike most of the micropayment protocols, user anonymity is supported, such feature is achieved by the elliptic curve digital signature.

In the second application, a new feature of receipt-free electronic voting system is proposed. In early electronic voting papers, a receipt was generated and returned to the voter so as to prevent any modification of a submitted ballot.

However, such receipt could be used as an evidence of the choice made by a voter, thus allowed the selling of votes. Because of this, the concept of receipt-free electronic voting was proposed by some researchers. However, we find out a serious security threat in most of the cryptography-based receipt-free electronic voting systems: the voting right is transferable. Therefore, we have designed a new cut-and-choose authentication algorithm, in which the private key is involved while in the initial phase the prover remains anonymous. This algorithm enables the non-transferability of voting right in a receipt-free electronic voting system, which is essential requirement in a fair voting.

In the third application, we proposed an anonymous Internet access services via proxy server, which is motivated by the electronic cash protocol. When a user access the Internet through the proxy server resides in the Internet Services Provider, information such as when and how long the user has accessed the web, which web sites have been visited and which Internet objects were requested etc are recorded. This enables one's Internet usage habit being analyzed and is a threat to the Internet privacy. Therefore we have designed a cryptographic application which provides conditional anonymity to the users.

These applications will be described in details in this thesis.

1.4 Organisation of the Thesis

The thesis is organized as follows. In Chapter 2, cryptographic primitives that are involved in the thesis are explained, they include *One-way Hush Function*, *Bit Commitment*, *Secret Splitting*, *Symmetric and Asymmetric Encryption*, *Blind Signature*, the *Cut-and-choose Protocol*, the *RSA Public Key Cryptosystem* and

the *Elliptic Curve Cryptosystem*. Chapter 3 gives a general study on Money. This includes the history of money, the functions of money as well as some existing payment systems. Through this chapter we can obtain a general view of money and its features, so that the protocols introduced in the later chapters can be more easily understood.

In Chapter 4 we discuss the electronic cash protocols. A basic model of electronic cash is described and several existing examples of electronic cash protocols are also introduced. In the last section of this chapter, we talk about the limitation of electronic cash protocols and the possible modifications.

Chapter 5 is a discussion on the micropayment protocols. We emphasis on the efficiency that is required for the micropayment systems. A basic micropayment model is described and several existing examples of micropayments are also given.

In chapter 6 to chapter 8, the three applications derived from the electronic cash protocols are described in details.

Two publications have been derived from this thesis. Namely *On Privacy Issues of Internet Access Services via Proxy Servers* that derived from chapter 7 and *Anonymous Electronic Voting System with Non-Transferable Voting Passes* that derived from chapter 8. The research result from chapter 6 may also be implemented.

Chapter 2

Cryptographic Primitives

Many of the procedures involved in the payment systems can be analog digitally. This is achieved by the cryptographic techniques. In this chapter, we explain the cryptographic primitives involved in the applications that are mentioned in this thesis.

2.1 One-way Hash Functions

A hash function h maps an input x to an output $h(x)$. It is called one-way because it is computationally infeasible to calculate x from $h(x)$.

A two-argument one-way hash function takes two input x, y and generate an output $h(x,y)$. It is computationally infeasible to calculate x and y from $h(x,y)$.

2.2 The Bit Commitment Protocol

Suppose Alice wants to commit to a prediction but does not want to reveal this prediction to Bob before a certain point of time. And Bob, at the same time, wants to ensure that Alice cannot change her mind once she has committed to the prediction. In such case, a bit commitment protocol can be employed.

The following steps describe how Alice commits on a bit b:

1. Alice generates two random bit strings R_1 and R_2.
2. Alice generate the following token:

$$(R_1, R_2, b)$$

3. The message is hashed using some suitable one-way hash functions $H(.)$. The follows are presented to Bob:

$$H(R_1, R_2, b), R_1$$

4. Later when Alice opens her committed bit, she submits the followings:

$$(R_1, R_2, b)$$

5. Bob hashes R_1 R_2, b using $H(.)$ and checks if the result is valid. If so, he accepts Alice's committed bit.

2.3 Secret Splitting

In the secret splitting scheme, a message is divided into pieces and each pieces give no information about the message [4]. When all pieces are put together, the message is recovered.

The algorithm is described by the following steps:

1. A random bit string R which is the same length as the message M is generated.

2. Obtain P by

$$P = M \oplus R$$

where $\oplus$ denotes the binary XOR operation.

3. The message is divided in to two pieces, P and R.

The message M is recovered when one put P and R together:

$$P \oplus R = M \oplus R \oplus R = M$$

2.4 Encryption/Decryption

Encryption is a process to transform a readable message (called the *plaintext*) into a format so that the original information can not be retrieved. An encryption key is used. The encrypted message is called the *ciphertext.*

Decryption is a reverse process of encryption. It is the transformation of the ciphertext back to the plaintext with a decryption key.

The encryption and decryption processes are described in the followings:

$$E_{ek}(P) = C$$
$$D_{dk}(C) = P$$

where P = the plaintext

C = the ciphertext

ek = the encryption key

dk = the decryption key

There are two kinds of encryptions: symmetric encryption and asymmetric encryption.

2.4.1 Symmetric Encryption

In symmetric encryption schemes, the encryption key is the same as the decryption key. Therefore, the communicating parties should firstly agree on the encryption/decryption key. Examples of symmetric encryption include the Data Encryption Standard (DES) [5] [6], Triple DES [7] and the International Data Encryption Algorithm (IDEA) [8].

2.4.2 Asymmetric Encryption

In asymmetric encryption schemes, the encryption key is different from the decryption key. The receiver publishes the encryption key in a public directory so that potential senders can use this key to encrypt the message. And the receiver uses the privately stored decryption key to decrypt the message. The encryption key is called the public key and the decryption key is called the private key.

Asymmetric encryption is also called the public key encryption. Examples include RSA [9] and Elliptic Curve Cryptosystem [11] [12].

2.5 The RSA Public Key Cryptosystem

The RSA public key cryptosystem [9] is very widely used in many cryptographic applications. It is named after its inventors, R. Rivest, A. Shamir, and L. Adleman. The security of RSA is based on the difficulty of factoring a very large number, which is the product of two large primes.

We describe the mechanism of RSA public-key encryption in the following:

To generate the RSA public and private key pairs:

1. Two large distinct primes p and q, each roughly the same size, are generated.

2. Compute $n = p \cdot q$.

3. Pick a large integer d such that

$$gcd(d, (p-1) \cdot (q-1)) = 1.$$

4. Compute the integer e such that

$$e \cdot d \equiv 1 \pmod{(p-1) \cdot (q-1)}.$$

5. The public key is (n, e), while the private key is d.

The signature s on a message M using the private key d is

$$s = M^d \pmod{n}.$$

To verify the signature, one checks if the following equality holds:

$$s^e = M \pmod{n}$$

The above equality holds if s is a signature on M using the private key d because

$$d \cdot e \equiv 1 \pmod{(p-1) \cdot (q-1)}$$

and by the Little's Fermat's Theorem,

$$s^e = (M^d)^e = M^{de} = M \pmod{n}.$$

2.6 Blind Signature

The blind signature technique was invented by David Chaum [1], which is based on the RSA algorithm.

Suppose Alice wishes to obtain a blind signature from Bob, who has a public key e, a private key d and a public modulus n. To obtain Bob's signature on the message M while giving Bob no information on M, the following steps are taken:

1. Alice choose a random integer r.

2. Calculate

$$T \equiv r^e \cdot M \pmod{n}.$$

3. Bob signs on T with his private key d:

$$T^d \equiv (r^e \cdot M)^d \pmod{n}$$

and returns to Alice.

4. Alice unblinds T^d by dividing T^d with r.

5. A signature on M is obtained because:

$$T^d/r \equiv (r^e \cdot M)^d/r \equiv M^d \pmod{n}$$

Bob cannot read M because it is blinded by r. In order to protect Bob from being cheated by Alice, a cut-and-choose protocol [10] can be used along with the blind signature. We describe the cut-and-choose protocol protocol in the following section:

2.7 Cut-and-choose procotol

Suppose Alice and Bob are dividing a cake, in order to obtain a fair result, the following steps are taken:

1. Alice cuts the thing in half.

2. Bob chooses one of the halves for himself.

3. Alice takes the remaining half.

In this way a fair result can be obtained. It is in Alice's best interest to divide fairly in the first step, because and Bob is able to choose whichever half he wants.

The cut-and-choose protocol can be applied along with the blind signature so as to protect Bob from being cheated by Alice [19]. This is described as follow:

1. Alice chooses k random integers r_i for $i = 1$ to k.

2. She calculates

$$T_i = {r_i}^e M \pmod{n} \text{ for } i = 1 \text{ to } k$$

And submit the k T_i to Bob.

3. Bob chooses an integer j randomly and asks Alice to show the values of r_i for all $i = 1$ to k except j.

4. Bob verifies if $T_i = {r_i}^e M \pmod{n}$ for $i = 1$ to k except j.

5. If the verification in (4) is passed, he signs on the unopened T_j's and returns it to Alice.

6. Alice unblinds ${T_j}^d$ by dividing ${T_j}^d$ with r_j.

7. A signature on M is obtained because:

$${T_j}^d / r_j \equiv ({r_j}^e \cdot M)^d / r_j \equiv M^d \pmod{n}.$$

The possibility of Alice being successful in cheating is $1/2^k$ since Bob choose j out of the k candidates randomly.

2.8 The Elliptic Curve Cryptosystem (ECC)

The Elliptic Curve Cryptosystem (ECC) is proposed by Neil Koblitz [11] and Victor Miller [12] in 1985 independently.

An elliptic curve is the set of solution *(x,y)* to an equation of the form:

$$y^2 = x^3 + ax + b \pmod{p} \tag{2.1}$$

for two integers a and b, and p can be a prime number or 2^m.

2.8.1 The Elliptic Curve Discrete Logarithm Problem

Suppose P and Q are both points on the curve described by (2.1), then $P + Q$ will always be another point on the curve [13]. xP represents the point P adding itself for x times.

Suppose Q is a multiple of P, so that

$$Q = xP$$

for some x. Then the elliptic curve discrete logarithm problem is to determine x given P and Q. In fact, the security of ECC is based on the difficulty of the elliptic curve discrete logarithm problem.

2.8.2 Cryptographic Applications Implemented by ECC

The multiplication operation in ECC is an analogy to the modular exponentiation. As mentioned in last section, the security of ECC is based on the difficulty of the elliptic curve discrete logarithm problem. Cryptographic applications such as key exchange algorithms, data encryption as well as digital signatures can be implemented with ECC. They are described in the following subsections.

2.8.3 Analog of Diffie-Hellman Key Exchange

The Diffie-Hellman key exchange algorithm can be implemented by ECC in the following way:

Preparation

1. Select the modulus p and the elliptic curve parameters a and b for the equation

$$E_p(a,b) : y^2 = x^3 + ax + b \pmod{p} \tag{2.2}$$

2. Pick a generator point G on $E_p(a,b)$ such that the smallest value n for which $nG = O$ is a very large prime number.

3. $E_p(a,b)$ and G are the parameters of the cryptosystem known to all participants.

Key Exchange

1. Alice selects an integer n_A (less than n) as her private key. The corresponding public key is $P_A = n_A G$.

2. Bob selects his private key nB and his public key P_B similarly.

3. Alice obtains the secret key K by calculating $K = n_A P_B$. Bob obtains K by calculating $K = n_A P_B$.

The secret key obtained by Alice and Bob are in fact the same key since

$$n_A P_B = n_A(n_B G) = n_B(n_A G) = n_A P_B.$$

2.8.4 Data Encryption [11]

Suppose Alice wants to encrypt and send to Bob a message P_m using the elliptic curve $E_p(a,b)$ defined in (2.2). Suppose Bob's private key and public key are n_B

and $P_B = n_B G$ respectively, where G is a point described above. The following steps are taken by Alice:

1. The message M is mapped onto a point P_m on $E_p(a, b)$.
2. A random integer k is selected.
3. Alice sends Bob the pair of points:

$$(kP_B, P_m + k(n_B P_B)).$$

To decrypt the message, Bob multiplies the first point in the pair by n_B and subtracts the result from the second point:

$$P_m + k(n_B P_B) - n_B(kP_B) = P_m$$

and obtain M by the inverse mapping of P_m.

2.8.5 The ECC Digital Signature

The digital signature algorithms can also be implemented by elliptic curves. And it will be described in section 4.3.

Chapter 3

What is Money?

The term *electronic money* was originated in Robert Hendrickson's *The Cashless Society* [14]. In his book, the pros and cons of electronic cash (in forms of credit card and electronic fund transfer) over traditional paper cash was widely discussed. However, the concept of currency that exists in format other then paper and coins has been expressed as early as 1888 in Edward Bellamy's book *Looking Backward* [15], in which cash was replaced by pasteboard and are hole-punched at each transaction.

In this chapter we give a description of the history as well as the properties of money.

3.1 Money

What is money? The following description is reference from an economist book *Introduction to Money* written by Honor Croome [16]:

> *Money is a store of 'liquid' value, a perfectly disposable asset. It is a medium of exchange, accepted by anyone in any transaction - that is what constitutes its liquidity. Its existence makes possible the keeping of accurate and meaningful economic records and the making of economic estimates; that it is constitutes a unit of account.*

When we talk about electronic money, we are referring to the substitution of *money* by any electronic formats. Examples of electronic money include credit cards and electronic checks. Electronic cash and micropayment are more specifically, the electronic analogy of cash. We are going to talk about their example in the following sections.

3.1.1 The History of Money [17]

In the early days of human history when money is not yet invented, people practiced barter: the direct exchange of goods and services for other goods and services. However, this transaction method requires a *double coincidence of wants*. Therefore, such practice is replaced by the usage of various forms of money.

The earliest money was the *commodity money*. It is an item that is accepted in trade. These physical commodities which was used as a medium of exchange include seashells, beads, tea, fish hooks, fur, cattle and even tobacco.

Later expensive metals such as silver and gold were used as money. This added portability to the money. In 2500 BC the Egyptians produced metal rings and used them as money. By 700 BC, a group of seafaring people called the Lydians became the first in the Western world to make coins. During the 18th

century, coins became popular throughout Europe as trading grew. However, the coins make by silver and gold are still belong to the group of commodity money because they hold actual value.

In modern monetary systems, the essential feature of any medium in which payments are made is not the intrinsic value, but the general acceptability. The appearance of representative money further enhanced the portability of money. *Representative money* is tokens or pieces of paper that are not intrinsically valuable themselves, but can be exchanged for a specific commodity. *Fiat money* is similar to representative money except that it cannot be redeemed for a commodity. The notes we use today are an example of fiat money.

As technology advances and more convenience is desired, electronic money begin to eliminate part of the physical cash as a transaction medium. Credit cards were introduced in the fifties and it is widely used in the world today. And a new form of money: *electronic cash* was developed in the nineties. Electronic cash enable Internet and wireless payments, which enhances the conduction of electronic commerce.

3.1.2 Functions of Money

In his book *A History Money* [18], Davies, Glyn presents the following definition for money:

> *Money is anything that is widely used for making payments and accounting for debts and credits.*

According to [18], there are general and specific functions of money. The general functions of money include:

1. As the liquid asset.

2. As a framework of the market allocation system, that is, the prices for goods and services.

3. As a causative factor in the economy.

4. As the controller of the economy.

The general functions of money are mostly macro-economic. There are also specific functions of money, which include the followings:

1. As a unit of account.

2. As a common measure of value.

3. As a medium of exchange.

4. As means of payment.

5. As a standard for deferred payments.

6. As a store of value.

The specific functions of money are mostly micro-economic.

Not every commodity used as money can achieve all the general and specific functions listed above. Furthermore, the functions of any particular form of money may change over time. Therefore, when a new form of money is developed, it is necessary for the society to consider if this new money can possess as many functions as listed.

3.2 Existing Payment Systems

Nowadays, the method of transaction settlements is not limited to the usage of coins and notes. Yet, payment by cash is the simplest form of payment. However, depends on the nature of payments, many different alternatives are used.

3.2.1 Cash Payments

Cash refers to coins and bank notes. Payment by cash is the simplest form of payment. It is easily transferable from one individual to another. And it is suitable for payments of small amount because there is no transaction charges levied. It is also anonymous because no audit trail is left behind.

However, there is some weakness of cash. For example, it is not portable when a large amount of cash is to be handled. Also, such large amount of cash has to be transferred with a very high level of security, thus results in a high handling cost. Also, the problem of counterfeits becomes more severe as photocopying technology advances.

3.2.2 Payment through Banks

Suppose Alice, who keeps her money in Bank A need to pay Bob, who has an account in Bank B for a certain sum of money. It may not be efficient for Alice to withdraw the money from Bank A and pass to Bob, who deposits it back in Bank B. Therefore, some other payment methods other than payment by cash are employed. These methods include *payment by check*, *payment by giro or credit transfer*, *automated clearing house (ACH) payments* and also the

wire transfer services. They are categorized as *payment through banks* since the banks are involved.

3.2.3 Using Payment Cards

In 1947, the Flatbush National Bank issued cards to its local customers. In 1950, the first charge card, *Travel and Entertainment* was issued by the Diners Club. Nowadays, there are two major companies, VISA International and Mastercard makes up a large number of member banks.

Traditionally, there are two kinds of card payment systems: the credit card payment systems and the debit card payment systems.

In **credit card payment systems**, the card-issuing banks register the cardholder, produce a card incorporating and the card association's logo, and operate a card account to which payments are charged. The merchants need to register to the banks too. When a cardholder purchases in a registered merchant, a sales voucher containing the payer's card number, the amount of the payment, the date and a good description. At the end of the day, the merchant presents the vouchers to the bank where the payments are cleared. The merchant's account is credited while the cardholder's account is debited.

The **debit card payment systems** ars similar, but the transaction clearance is done real time. That is, at the time the transaction takes place, the amount is transferred from the customer to the merchant bank account.

In the recently years, smart card technology develops rapidly and smart cards begin to join the market. They will be discussed in the next chapter.

Chapter 4

Electronic Cash

Electronic cash was the digital analogy of cash. The classical electronic cash protocol was proposed by David Chaum et al. in [19]. It is the first electronic cash protocol that can truly implement the features of paper cash.

4.1 The Basic Requirements

Since electronic cash is a digital implementation of paper cash, its requirements are very similar to those of paper cash.

According to [20], an ideal cash payment system should possess the following requirements:

1. **Independence**

   The security of electronic cash can only depend on itself, it cannot depend on any physical condition.

2. **Security**

   The cash can neither be copied, forged or double spent.

3. **Privacy**

   User privacy should be protected. No one should be able to trace the relationship between a user and his purchases. In the other words, user anonymity should be provided. This requirement is also referred as the *untraceability* requirement.

4. **Off-line payment** During a transaction, the only parties involved are the payer and the payee. No customer-bank or merchant-bank on-line connection is required.

5. **Transferability**

   The cash can be transferred from one user to another.

6. **Divisibility** The piece of cash can be subdivided into arbitrary pieces of smaller values so that the value of all pieces is equivalent to the value of the undivided piece of cash.

To achieve the above requirements, cryptographic algorithms such as blind signature, zero-knowledge proofs and public-key signatures are employed. We describe the basic electronic cash model in next subsection.

4.2 Basic Model of Electronic Cash

In this section, we describe the basic model of electronic cash proposed in [19]. [19] is chosen because it is a significant paper in the history of electronic cash and the three applications in this thesis: *Micropayment Tickets*, *Anonymous Electronic Voting* as well as *Anonymous Internet Access* are motivated by this paper.

In [19], a electronic coin exists in the format of

$$(x, f(x)^{1/3} \pmod{n})$$

where the first term represents the money and the second term is the bank's signature on the money.

We describe how Alice who has an account in an electronic cash issuing bank withdraws electronic coins and spend it in the merchant Bob. Let the private key and public key of the bank be d and e respectively, and n is the published composite in the public key cryptosystem.

4.2.1 Basic Protocol

We firstly present the basic electronic cash protocol. Where a modified version is given later.

1. Alice chooses two random numbers x and r. And presents

$$T = r^e \cdot f(x) \pmod{n}$$

to the bank.

2. The bank returns

$$T^d = (r^e \cdot f(x))^d \pmod{n}$$

that is

$$(r \cdot f(x))^d \pmod{n}$$

to Alice and withdraws the equivalent amount of dollar from Alice's account. Notice that $e \cdot d \equiv 1 \pmod{\phi(n)}$ as explained in chapter 2.

3. Alice removes the blinding factor r from T^d and form

$$(f(x))^d \pmod{n}$$

which is the bank's signature on x.

4. Alice can then form the electronic coin

$$C = (x, f(x)^{1/3} \pmod{n})$$

5. To spend one electronic coin in Bob's shop, Alice pays Bob C. Bob contacts the bank immediately and verifies if the coin has been deposited before. If not, Bob redeems the coin and his account is credited.

4.2.2 Modified Protocol

The basic protocol above illustrates a simplified electronic cash model. However, modifications are required because Alice's anonymity is protected unconditionally, which is unfair to the merchants. Alice may double spend without penalty. Also, it is an on-line system which requires more resources. In [19], modifications are made so that the system is off-line and only conditional anonymity is provided to Alice. In this way, double spending is prevented.

The modified electronic cash protocol consists of three phases, namely

1. The cash withdrawal phase
2. The spending phase
3. The redeeming phase

Cash Withdrawal

Assume Alice has a bank account numbered u and the bank records an associated counter v. To withdraw a coin from the bank, the following procedures are undergone:

1. Alice forms and sends k T_i's in the following manner:

$$T_i = r_i^3 \cdot f(x_i, y_i) \pmod{n}$$

where

$$i = 1 \text{ to } k$$
$$x_i = g(a_i, c_i)$$

and

$$y_i = g(a_i \oplus (u \parallel (v+i)), d_i).$$

where $\oplus$ denotes bitwise exclusive OR and $\parallel$ denotes concatenation.

2. The bank chooses randomly a set of $k/2$ integers, $R = \{i_j\}$, where $1 \leq i_j \leq k$ and $1 \leq j \leq k/2$. and asks Alice to show the values of r_i, a_i, c_i and d_i for every i in R.

3. The bank compares the $k/2$ presented T_i's and see if they can be derived from these r_i, a_i, c_i, d_i, u and v.

4. The bank gives Alice

$$\prod_{i \notin R} T_i^{1/3} \pmod{n}$$

and withdraws an equivalent amount of money from Alice's account. The value of the counter is also increased from v to $v+1$.

5. Alice removes the blinding factor and form

$$\prod_{i \notin R} f(x_i, y_i)^{1/3} \pmod{n}$$

which is corresponding to the bank's blind signature on the electronic coin

$$C = \prod_{i \notin R} f(x_i, y_i) \pmod{n}.$$

Alice also reindexes the candidates in C in a lexicographic sense such that:

$$f(x_1, y_1) < f(x_2, y_2) < \ldots f(x_{k/2}, y_{k/2}).$$

she saves the electronic coin and also increases her copy of the counter by 1.

The Spending Phase

To spend one coin in Bob's shop, the following procedures are run:

1. Alice sends Bob the coin C.

2. Bob generates a $k/2$-bit binary random challenge vector $\mathbf{z} = (z_1, z_2, ..., z_{k/2})$ and sends it to Alice.

3. Alice responds according to the following rule:
   - when $z_i = 1$, Alice sends Bob a_i, c_i and y_i;
   - when $z_i = 0$, Alice sends Bob $x_i, a_i \oplus (u \parallel (v+i))$ and d_i.

4. Bob verifies that C can be derived from these partial openings presented by Alice.

The redeeming phase

Bob sends C along with the partial openings to Alice's bank and his account is credited.

The bank also stores C, together with $\mathbf{z}$ and the corresponding partial openings.

The modified protocol supports an off-line payment system because Bob need not to contact the bank immediately after receiving Alice's payment.

4.2.3 Double Spending Prevention

In the modified version of electronic cash protocol, double spending is prevented. This is because if Alice spends a same coin at two different merchants, two random challenges on her coin will be executed. Suppose she spends the coin firstly at M_1 and then at M_2, each generates the random challenge vectors $\mathbf{z_1}$ and $\mathbf{z_2}$, where:

$$\mathbf{z_1} = (z_{1_1}, z_{1_2}, ..., z_{1_{k/2}})$$

$$\mathbf{z_2} = (z_{2_1}, z_{2_2}, ..., z_{2_{k/2}})$$

When M_2 redeems the coin, the bank will discover that Ailce double spent. Suppose $z_{1_i} \neq z_{2_i}$, the bank will receive the following partial openings:

$$a_i, c_i, y_i \text{ and}$$

$$x_i, a_i \oplus (u \parallel (v+i)), d_i$$

and Alice's account number u will be revoked since

$$a_i \oplus (a_i \oplus (u \parallel (v+i))) = (u \parallel (v+i))$$

and the bank has knowledge on $(v+i)$ once u is known.

The chance of Alice being undetected in a double spending is $1/2^{k/2}$.

4.3 Examples of Electronic Cash

There are several examples of electronic cash that come into real applications. We describe them briefly in this section.

4.3.1 eCash

eCash [19] is founded by David Chaum et al and its protocol is described in details in last section. It is now sold to the **eCash Technologies** [22], an US based company. Advanced electronic cash technologies such as person-to-person electronic payments and mobile electronic cash are being developed. eCash is a tokenized cash system because the coin represents actual monetary value.

4.3.2 CAFE

CAFE (Conditional Access for Europe) was a project funded under the European Community's ESPRIT program [23], [24]. The CAFE protocol is based on the idea of the untraceable electronic payment protocol proposed by David Chaum in [25] and the concept of checks with counters [26]. It is a hybrid scheme in which it offers all the benefits of anonymous electronic cash but at the same time allows the users to sign up checks up to a specified amount. Strong cryptographic techniques and temper-resistant devices are used so that all the

payment transactions are protected.

4.3.3 NetCash

Netcash [27] is a framework supporting anonymous real-time electronic payment systems over an opened network. It provides scalability and acceptability with weaker anonymity and only a limited form of offline-operation. Moreover, according to [27], it integrates anonymous electronic currency into the global banking and accounting infrastructure.

In the NetCash system, there are several currency servers. A user buys electornic coins from these currency servers and spends it in a merchant. The merchant checks from the currency servers whether the electronic coins are redeemed before. An unredeemed electornic coin can circulate until it is redeemed at the currency servers.

4.3.4 CyberCash

CyberCash [28] is an on-line cash-like system. The system is designed for those payments where the value is too low to be paid by credit cash transactions. Customers buy CyberCoin cash from the CyberCash server, and the amount is charged to their credit cards or bank accounts. The cash is then stored in a CyberCash wallet, which is a piece of software resides on the customer's side.

In CyberCash system, the electronic coin metaphor is a lot weaker then the tokenized systems such as eCash. An account is established with the CyberCash server when a customer buys CyberCoin cash. To make a payment is actually similar to authorizing a transaction of an amount from one account to

another. Public-key cryptography is used when loading a CyberCash wallet, while symmetric key is used during CyberCoin transaction.

4.3.5 Mondex

The **Mondex** electronic cash card [3] is referred as a *prepayment card.* There are very little publicity about the Mondex transaction algorithm. According to [3], the Mondex electronic cash system operates on a smart card which stores value on a microchip. Complex security systems are used when value is transferred between chips.

The Mondex Card is a smart card with a small microcomputer chip embedded in it, which acts as an electronic purse. The electronic purse can be loaded with value, and the value is stored inside the smart card until it is used as payment for goods or services, or transferred to another Mondex Card. Payments are done by inserting the Card into a card reader. The electronic purse can also be locked using a personal code so that only the Card's owner can access the value on it.

4.4 Limitations of Electronic Cash

Since electronic cash carries monetary value, it should be very carefully designed. Therefore the security requirements are strict and the computational effort required is high. Similar to credit card payments, the cost of transaction of electronic cash becomes significant when the amount involved is low. Therefore electronic cash is not suitable for low-value transactions.

Because of the above reason, payment protocols that are suitable for low

amount transactions are designed. They are referred as micropayment protocols. And they will be discussed in details in next chapter.

Chapter 5

Micropayments

In electronic cash protocols, a significant amount of public-key cryptosystems is employed. This makes them secure but at the same time also increase the transaction cost. As we enter into the digital age, new electronic commerce applications are developed and there are new requirements for payments. For example, the payment for an on-line stock quote as well as the payment for viewing an article on web. Therefore, an efficient digital payment system that can support the payment of a very small value is needed.

Micropayments are electronic payments of small amount, which can be as little as a penny. Since the amount involved is small, both high computational and storage efficiencies are required so that the transaction cost becomes insignificant. This can be achieved by giving up some of the security requirements such as user anonymity; or making the payment low-value itself such that the cost of counterfeiting is higher than the micropayment itself.

In order to distinguish from the *micropayment* schemes discuss this chapter, we refer the electronic cash schemes mentioned in last chapter as *macropayment*

schemes.

5.1 Basic Model of Micropayments

Exceptional efficiency is required in order to support micropayments. For example, in [29], hash operations are used to minimize the number of public-key operations. According to [29], a hash function evaluation is about 100 times faster than RSA signature verification, and is about 10,000 times faster than RSA signature generation. Therefore hash functions are usually employed in micropayment schemes. Table 5.1 gives a comparison between the computational speed of hash functions and the other operations.

Operation	**Number per second**
One-way hash function evaluation	20000
Public-key signature	2
Public-key signature verification	200
Network connection	1000

Table 5.1: Comparison of Computational Speed of Hash Functions Evaluation against other Operations

Similar to electronic cash, three parties are involved in a micropayment model, namely the broker, the user and the vendor:

1. **User:** the customer who obtains goods and services by paying with micropayments

2. **Broker:** minter of micropayments

3. **Vendor:** the merchant who provides goods and services in exchange for micropayments

5.1.1 Micropayments generation

There are two ways to generate the micropayments: either generated by the brokers or generated by the customer.

Generated by the brokers

In this method, the broker mints micropayments. And customers buy micropayments from the brokers with any other macropayment methods. A return policy is also required so that the customers can return any unused or expired micropayments. Such scheme is a debit-based scheme since the customer has to pay before they obtain the good and services.

Generated by the customer

In this method, a certified customer generates the micropayments by himself and spends them in he vendors. His account will be debited when the vendors redeem the micropayments at his bank. This scheme is credit-based because the customer can spend before he actually pays.

5.1.2 Spending

Unlike the macropayment schemes, secure one-way hash functions, instead of public-key signatures, are used in the micropayment schemes. A micropayment chain, or the stack of coins M has the following format (Assume the stack consists

of 100 coins):

$$M = (N, f^{100}(N), Sign(C, f^{100}(N)))$$

where $Sign$ denotes the signature of the coin issuer and C denotes the certificate of the stack owner. $f(.)$ is a secure one-way hash function, $f^{i+1}(x) = f^i(f(x))$, $f^0(x) = x$.

To spend i coins from an unspent chain, the customer sends the following to the vendor:

$$(f^{100}(N), f^{100-i}(N), Sign(C, f^{100}(N), i)$$

The vendor verifies if $f^i(f^{100-i}(N)) = f^{100}(N)$, as well as other information. Upon successful verification, the vendor accepts the payment. The unspent stack is either return to the broker or authorized by the vendor.

Double spending is a common problem in all electronic payment schemes, including both macro- and micropayments. Different measures are used in different schemes to prevent double spending. Usually a database is used to lookup all payment protocols so that the double spender is detected. In this way double spending is deterred.

5.1.3 Redemption

In micropayment schemes, an offline redemption process is required because of the computational power consideration. That is, the vendors verify and redeem the micropayment collected later in the end of the day. A record of redeemed micropayments is kept in the database of the broker so as to make sure that the micropayments are only redeemed once.

5.2 Examples of Micropayments

Micropayment protocols are developed since mid nineties. In this section, we describe three representative micropayment schemes. Namely the Payword[29], MicroMint[29] and Millicents[30].

5.2.1 PayWord

PayWord is a credit-based micropayment scheme. It uses chains of hash values described in section 5.1.2 to represent customer credit within the system.

The paywords are generated by the customers. A customer picks up a random number w_n as the n^{th} payword. It is also used as the seed for generating the rest of the payword chain in the following way:

$$w_{i-1} = h(w_i) \text{ where } i = 1 \text{ to } n$$

and $h(.)$ is a secure one-way collision resistant hash function. The zeroth value, w_0 is called the root of the chain and is included in the commitment M, where

$$M = Sign_{broker}(w_0, vendor, customer, date, other\ information).$$

As indicated from M, the payword chain is bounded to a particular customer-vendor relationship by the above commitment. That is, the customer can only spend the chain in a specific vendor once the commitment is signed.

To make a payment, the customer sends

$$P = (w_i, i)$$

along with the commitment M to the vendor. The vendor verifies if w_i equals $h^i(w_0)$. If so, it accepts the payment. It also records w_i in a database as a mark

of the last spent payword. The unfurnished payword chain can be used again later. While $P' = (w_{i+k}, k)$ will be sent when the customer spends the next k coins.

When a vendor redeems the coins, the last payword and the customer's commitment are sent to the broker. Upon successful verification, the vendor's account is credited while at the same time the customer's account is debited.

5.2.2 MicroMint

MicroMint is proposed at the same time as PayWord [29]. It is debit-based to the customers and the broker, but credit-based to the vendors. Its security relies on the difficulty to produce the coins without special cryptographic hardware resides at the broker's side. The MicroMint coins are produced by the broker, which are sold to the users who spend the coins at the vendors in exchange for goods and services. The vendors redeem the coins at the broker later.

The coins in the MicroMint protocol are specially designed. It requires high generation cost unless the coins are generated in bulk. Special cryptographic hardware is used and economy of scale is achieved at the broker side.

The MicroMint coins are k-way-collision tuples $(x_1, \ldots, x_k)$, where the x_i's are distinct and

$$h(x_1) = h(x_2) = \ldots = h(x_k) = y$$

for some number y. [29] gives an efficiency analysis in producing the coins. How the coins are spent, verified and redeemed are similar to other electronic payment protocols.

5.2.3 Millicent

Millicent is developed by Digital Equipment Corporation in 1995 [30]. It is designed for the payment as low as at tenth of a cent. It is a debit-based system and the Millicent coins are called *script*.

Millicent is a decentralized system. The vendors issue scripts while the broker buys these scripts from the vendors in bulk. Before a customer purchases goods and services from the vendors, he buys the corresponding scripts from the broker. To make a payment, the followings are sent from the customer to the vendor:

$$(script_{vendor}, request, H(script_{vendor}, request, CSK))$$

where CSK is the customer shared key which is unique for every customer. $H(.)$ is a one-way hash function. To prevent double spending. The vendor checks the database to see if the script is spent before. When necessary, a change $script'_{vendor}$ will be given back to the customer.

There is no redemption in Millicent because the vendors are paid already when the broker buys the scripts. A multiple brokers environment is also support [30].

5.3 Limitations of Micropayments

Micropayments are specially designed for making transactions of low values. Because of its functions, micropayment protocols are efficient in terms of computational power and storage requirement. However, in order to achieve these efficiencies, the security requirements are loosen.

While macropayment systems are on-line and they are more concerned with authenticity and privacy. Some relatively complicated cryptographic algorithms such as public key signature as well as zero-knowledge proof of identity are involved. Micropayment protocols, on the other hand, use simpler algorithms such as symmetric encryption and hash functions to achieve a limited security. However, if the micropayment protocol is carefully designed, adequate security can also be supported. For example, in Chapter 6 we introduced the *Anonymous Micropayment Ticket*, which is a micropayments scheme with conditional user anonymity.

5.4 Digital Money - More than a Medium of Transaction

In Chapter 4 and 5, we have introduced a number of digital money protocols, which are constructed from different combinations of cryptographic primitives. Based on these existing digital money protocols, new applications such as electronic voting systems [31] [32], anonymity services provision [33] [34] as well as electronic auction systems [35] are developed.

In this thesis, three cryptographic applications are developed based on the electronic cash protocols. Namely:

Anonymous Micropayment Tickets

This is an anonymous micropayment protocol. Traditional micropayment protocols give up user anonymity for computational efficiency [29] [30]. Our protocol introduces a new concept of payment tickets, which can be considered as a

reusable electronic coin. With the employment of elliptic curve digital signatures [36], double spending is prevented. The system will be explained in Chapter 6.

Non-transferable Electronic Voting Pass

The security requirements in an electronic voting system are similar to those in electronic cash systems. In an electronic voting system, each legitimate voter is given a 1 dollar electronic coin and they use the electornic coin to 'buy' their preferred candidates during the polling period. After the polling period, the candidate who gets most coins wins. Similar to the electronic cash systems, desirable properties such as voter anonymity and double-voting prevention are achieved.

In this thesis, a receipt-free electronic voting system is proposed. In our proposal, we have modified the coin-withdrawal algorithm in the electrnic cash protocol and achieved the non-transferability of voting right. This protocol will be described in Chapter 7.

Anonymous Internet Access

When one accesses to the Internet via an Internet services provider (ISP), information such as when and how long one has accessed the modem pool and which objects one has requested is logged in the proxy servers. Such data enables one's user habit to be traced and analyzed. This is referred as 'clicktrails' data collections and is a threat to user privacy.

In this thesis, we proposed an anonymous Internet access service solution, which is based on cryptographic techniques. We have modified the electronic cash protocol in [19]. This solution enables the provision of Internet user

anonymity and the problem of 'clicktrails' data collection can be prevented. This will be introduced in Chapter 9.

Chapter 6

Anonymous Micropayment Tickets

6.1 Introduction

Electronic commerce has already become part of our daily life, in which the nature of business activities varies from ordering of physical goods to subscription of information. To support the transactions involve in the digital economy, two main types of electronic payment schemes: macropayments and micropayments are developed. As explained in chapter 4 and chapter 5, the former is for transactions of a larger sum of money, while the later is for transferring a smaller amount.

Since the amount of money involved in micropayment systems is small, in order to be cost effective, such systems need to be more efficient and lighter in weight when compare to the macropayment systems. In this chapter, we present an off-line credit based micropayment ticket, which can be used for

more than once. Also, the ticket is non vendor-specific, that is, a user can use the same ticket in different vendors. Moreover, our system supports conditional anonymity and we utilize a special feature of the elliptic curve digital signature so that the user identity is revealed when double spending occurs.

We firstly give a verbal description of the real life analogy of our system in section 2, followed by a review on elliptic curve digital signature in section 3. After that we will give a detailed description of the protocol in section 4, which is the cryptographic implementation for concepts introduced in section 2. In section 5 we will analyze on the security of our system. Efficiency analysis will be done in section 6.

6.2 Overview of the Systems

In this section, we give a real life analogy of the proposed system. Four parties are involved here, namely the user who purchases goods and services, the vendors who supply goods and services, the broker who mints micropayment tickets, and a TTP (trusted third party) that holds information about user identity. Both the user and the vendors are registered to the broker.

In our system, the user firstly obtains an anonymous certificate from the TTP. After that he/she opens an account in the broker with the anonymous certificate and buys micropayment tickets from the broker, which has the format shown in Figure 6.1.

The ticket consists of a table of 100 slots (this number is only for illustration and is arbitrary). It also contains some other information such as its serial number, the broker's signature as well as optional fields such as the value represented

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00
Serial number Broker's signature Other information ...																			

Figure 6.1: The micropayment ticket

by each slot and the expired date of the tickets etc.

To make a payment, the user presents his/her ticket to the vendor, after verification, the vendor records down the serial number of the ticket. It also crosses out x unmarked slots (where x equals *amount of payment divided by the value each slot represents*), starting with the smallest slot number and records down which slots it has crossed out. The ticket is then returned to the user. After this the user can use it at another vendor so long as the ticket has not been used up or expired.

To redeem the received payment, the vendor presents the information it recorded from the user to the broker. The broker checks if the corresponding slots on the ticket have been redeemed before. Upon satisfactory validation, the vendor's account is credited.

Since the ticket doesn't contain user information and is bought from the broker anonymously, our system is anonymous. That is, the broker cannot relate the identity of the user to the vendors whom he/she has been transacted, nor the vendor can derive the identity from the ticket. However, the anonymity provided in our protocol is conditional, in which the user identity is revealed when double spending occurs. This can be achieved by special feature of the

elliptic curve digital signature, which will be explained in next section.

6.3 Elliptic Curve Digital Signature

Since elliptic curve digital signature is a one-time signature and is defined in the IEEE standard [36], it is employed in our protocol to prevent double spending. Here we give a brief review of the signature scheme.

Firstly select a value q such that q is a prime number or $q = 2^m$, where m is a prime number. Also select an elliptic curve $E(F_q)$ over F_q (as described by *Equation 2.1*) and choose a point G of a prime order r in $E(F_q)$. Next select a random s as the signer's private key, such that $0 < s < r$. The corresponding public key is $W = sG$.

Before signing a message, a one-time key pair $(u, V = uG)$ is generated. Denote the point V as (x_v, y_v). The signature of a message m, where $0 \leq m < r$, is a pair of integers (c, d). Where

$$c = x_v \pmod{r} \tag{6.1}$$

$$d = u^{-1}(m + sc) \pmod{r} \tag{6.2}$$

To verify the signature, one computes $P = md^{-1}G + cd^{-1}W$. Denote P as (x_p, y_p) and the signature is valid if $x_p = c \pmod{r}$.

The one-time key pair $(u, U = uG)$ should be selected for every signature. Otherwise, the signer's private key can be recovered easily. For two message m_1 and m_2 signing with same u, the corresponding d_1 and d_2 are given by:

$$d_1 = u^{-1}(m_1 + sc) \pmod{r} \tag{6.3}$$

$$d_2 = u^{-1}(m_1 + sc) \pmod{r} \tag{6.4}$$

From (6.3) and (6.4), u is recovered by

$$u = (m_1 - m_2)/(d_1 - d_2) \pmod{r} \tag{6.5}$$

Once u is known, the private key s can be recovered by

$$s = (d_1 u - m_1)/c \pmod{r}.$$

In our protocol, payment is made by signing with the one-time key pair. Double spending is prevented because the private key is revealed when one uses the one-time key pair twice.

6.4 The Micropayment Ticket Protocol

In this section, we describe the cryptographic implementation of the micropayment tickets. Before Alice opens an account in the broker, she obtains an anonymous certificate *Cert* from the TTP:

$$Cert = Sign_{TTPsign}(pseudo,\ public\ key,\ expiry\ date)$$

where *pseudo* is the pseudonym of Alice. The TTP can relate *pseudo* to Alice but it reveals *pseudo*'s identity only when one can present the private key of *pseudo*. This condition has also been employed in the anonymous micropayment

system in [37]. However, it is not necessary for the TTP to know Alice's private key. When double spending occurs, this private key is revealed. Therefore the anonymity provided by the micropayment ticket is conditional.

With the anonymous certificate, Alice can opens an account in the broker using the name *pseudo*. She can obtain micropayment tickets from the broker. The payment can be made by any anonymous payment methods. Our system is credit based, where the account is debited only when the vendors redeem the payments.

6.4.1 The Micropayment Ticket

In this subsection we describe the actual structure of the micropayment ticket. A payword chain [29] is included in each ticket T. This chain consists of $w_0, w_1, ..., w_{100}$ where $w_i = h(w_i + 1)$ and $h(.)$ is a one-way hash function. Alice obtains T from the broker, where

$$T = Sign_{Brokersign}(w_0, w_1, ..., w_{100}, \textit{ pseudo, public key, expiry date}).$$

Along with the unused ticket, the broker also sends a spending token S_0 to Alice, where

$$S_0 = Sign_{Brokersign}(c_0, w_0).$$

S is an indicator on the status of the ticket. The first term c_0 is the x-coordinate of $U_0 \pmod r$ in the one-time key pair (u_0, U_0) chosen by Alice, while the second term is the last payword spent (w_0 in this case). Notice that u_0 cannot be derived from U_0. Also notice that Alice is anonymous to the broker and she can bought more than one ticket with the same account.

6.4.2 Payment

When *pseudo* (i.e. Alice) makes a payment to the first vendor V_1. Suppose she pays for a value worth n slots, she presents

$$P_1 = (T, Sign_{pseudosign}(spending\ information),\ S_0, Cert)$$

to V_1. The term *spending information* contains information that will be presented to the broker when the vendor redeems the payment, such as the first and the last slot numbers that involved in this payment (which is 1 and n in this case), the name of the vendor, time of transaction etc. The signature *pseudo sign* is an elliptic curve digital signature as described in last section, using the one-time key which is authorized by the broker who signed on S_0.

When V_1 receives P_1, it reads c_0 and w_0 from S_0. Next it verifies the signature on spending information with the public key shown on *Cert*. It also checks if the paywords on the ticket are valid by applying a hash function on the w_n n times and see if the result equals w_0. After this it saves T, $Sign_{pseudosign}(spending\ information)$, as well as S_0 for future redeeming. It also returns

$$S_1 = Sign_{V_1 sign}(c_1, w_i,\ spending\ information)$$

to *pseudo*, where c_1 is corresponding to the new one-time key pair chosen by *pseudo* and w_i is the last payword spent. This is the cryptographic analogy of the process of validating the ticket and crossing out the spent slot on it.

In general, pseudo presents

$$P_i = (T, Sign_{pseudosign}(i^{th}\ spending\ information),\ S_{i-1}, Cert)$$

to the i^{th} vendor, where S_{i-1} is the spending token signed by the $(i-1)^{th}$ vendor. In fact, it is not necessary for *pseudo* to send the entire ticket to the vendor. For

example, instead of showing all the values of w_0 to w_{100}, only the last payword in the payment chain are required.

6.4.3 Redemption

The vendor redeems *pseudo*'s payment by presenting T, $Sign_{pseudosign}$(spending information) and the spending token S to the broker. After verifying on the signature of *pseudo*, and knowing that the chain of redeeming paywords on the ticket are not redeemed before, the broker pays the vendor according to the amount listed in spending information. The account of *pseudo* will also be debited accordingly. Notice that the redemption of payword chains need not be in sequence.

6.4.4 Double Spending

Alice cannot spend a new chain of paywords without the spending tokens which marked the slot number of last spent payword. Therefore, if Alice double spends, she can only begin with those paywords which immediately after the end of any previously spent chains. This means that she has to use the same one-time key pair to sign on two different *spending information*, which are unique for every payment. From the explanation in section 3, her secret key can be recovered, and her identity is revealed by the TTP.

6.5 Security Analysis

In this section we analyze on the security of the anonymous micropayment ticket.

6.5.1 Conditional Anonymity

Anonymity is provided in our system. If the user doesn't double spend, the broker doesn't have information on the user's identity and the vendors cannot derive the user identity from the tickets. In this way, user privacy is protected. However, the anonymity cannot be abused because it is conditional and the user's identity is revealed when one double spends.

6.5.2 Lost Tickets

Only the owner of the ticket is able to spend the paywords on it. This is because every payment requires a signature of the ticket owner. The ticket owner bears no risk when he loses a ticket because the unused paywords cannot be spent by other people.

6.5.3 Double Spending

Double spending is prevented because the user's identity is revealed once the vendor redeems a double spent slot.

6.5.4 Collusion with Vendors

The user may cheat with a dishonest vendor by asking the vendor to sign on a false spending token. However this action can be detected easily when later the vendors redeem the payments.

Case 1

Suppose the following chains of payments which is shown in Figure 6.2 are redeemed, where the shaded part represents a forge chain resulted by the spending token signed by the dishonest vendor:

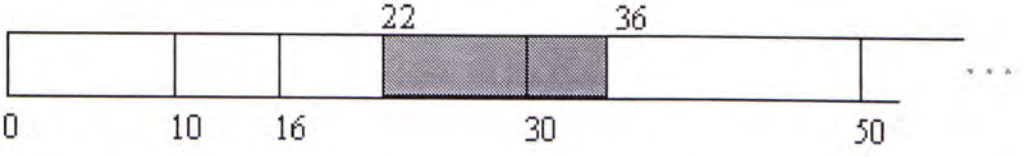

Figure 6.2: Double spending resulted from a forge spending token

This can be easily detected because the broker only trusts on the sequence of consecutive chains with which the first chain begins with w_0. This is because it must trust on the signer of S_0, that is the broker itself. In this way, the signer of the forge spending token is discovered easily.

Case 2

The dishonest vendor may also sign on a spending token S_x that contains a valid w_i (that is i equals a value at the end of a valid chain) and a new c, such that the user can use the paywords with slot number after $i+1$ twice. However they cannot success in cheating because when later the two vendors redeem the chains that both start with slot number $i+1$, two spending tokens both authorize slot number $i+1$ will be presented, which implies one of the spending tokens is forge. The dishonest vendor will be discovered because it is not the redeemer of the preceding chain.

Notice that the sequence of redemption of payword chains doesn't affect the ability of distinguishing dishonest vendors.

Traceable Consumption Habit

In our system, the vendors know the preceding vendors where a user has visited, so they can obtain limited information about the consumption habit of their customers. However it requires the collusion of a large number of vendors in order to trace the consumption habit of a user. Moreover, our system is anonymous in nature, if the user does not reveal his/her identity to the vendors, he/she still remains anonymous in the system under normal conditions. In this way his/her consumption habit cannot be traced.

6.6 Efficiency Analysis

The efficiency of our system relies on the repeating usage of a same micropayment ticket, which save much computation effort for generating new electronic coins. In this section we discuss the efforts by the TTP, the user, vendors and the broker.

The TTP's role in our system is a certification authority. It is responsible for issuing anonymous certificates and managing a secure database that holds the files about user identity. When in case of double spending, it needs to verify on the presented public-private key pair and reveals the user's identity when necessary.

The user needs to apply for an anonymous certificate and open an account in the broker. For each payment, an elliptic curve signature is made and the next one-time key pair is generated. He/she also needs to verify the vendor's signature on the new spending token.

A vendor needs to register to the broker. During each transaction, it needs

to verify three signatures (the broker's signature on the ticket, the previous vendor's signature on the spending token as well as the user's signature on the spending information) and check the validity of the paywords by performing a hashing n times. It also needs to sign on the spending token at the end of each transaction.

The brokers need to maintain the anonymous accounts and issuing micropayment tickets. When a vendor redeems the payments, the broker needs to verify the signature on the spending token, as well as that on the spending information. It checks if cheats occur and takes suitable actions when necessary.

6.7 Conclusion

To conclude, the anonymous micropayment ticket is a electronic payment that supports conditional anonymity, which is very important to user privacy but often being given up in micropayment systems. It is efficient and off-line. Moreover, the users are only liable to the payments spent with their signatures. Double spending is also prevented by the employment of elliptic curve digital signatures, which is well defined in the standard.

Chapter 7

Anonymous Electronic Voting Systems

7.1 Introduction

Since the first proposal of the cryptographic voting protocol in 1981 [38], a number of researches have been done on the electronic voting systems. Like many digital services, electronic voting brings much convenience to the society and lowers the cost of traditional elections. However, the systems must be supported by very strong security services in order to be fair and accepted by the general public. Efficiency is also an important factor for a practical system.

The definition of a secure voting system is given in [39], in which seven requirements namely completeness, soundness, privacy, unreusability, eligibility, fairness and verifiability are listed. However most of the early electronic voting schemes are not receipt-free. The receipt generated by the election administrator can be used as an evidence of having made a particular vote and therefore enables

one to sell his vote to the others. This problem induced the concept of receipt-freeness in an electronic voting system. Such a concept, which is firstly proposed in [40] is regarded as a milestone in electronic voting protocol design. A number of receipt-free electronic voting schemes have been proposed in which the receipts were eliminated by different means. In [40] and [41], physical assumptions are made so that receipt-freeness is achieved. While in [31], [42], [43] and [44], the authentication and voting processes are separated. A piece of authentication document is generated with which the voter can vote anonymously. However, we find out that in these systems, the voting right can be transferred because the authentication document is not related to any secret information of the voters. This is a severe threat to the fairness of a voting system.

7.2 The Proposed Electronic Voting System

7.2.1 The Proposed Election Model

Our system is designed for realising the digitalization of large-scale elections conducted by a government. Arbitrary voting schemes are allowed. It is assumed that the community is supported by a Public Key Infrastructure (PKI), where each voter possesses a public-private key pair and one can generate a liable digital signature with his own private key. The parties involved include the Voters, the Registrar, Administrator, Collector and a trusted third party (TTP).

1. **Voters**

   They refer to the group of people who have the right to vote in a particular election.

2. **Registrar**

   It is the server responsible for voter verification and anonymity service provision.

3. **Administrator**

   It is the server responsible for conducting the polling process and announcing the results.

4. **Collector**

   It is the server responsible for collecting the ballots. It is the digital analogy of the vote-collecting box.

5. **Trust Third Party (TTP)**

   This is an independent party who is responsible for handling disputes. The court is an example.

All of the Registrar, Administrator and the Collector are servers reside in the government or any parties that is responsible for holding the election. The Registrar and the Administrator may not necessarily be two separated servers. There is also a public counter, which is universally readable. The number shown represents the number of ballots accepted and its initial value is set to zero.

There are three phases in the election. The Registration Phase, the Polling Phase and also the Vote Opening Phase.

1. **The Registration Phase**

   In this phase, the valid voters registrar for the election and obtain an anonymous voting pass from the Registrar.

2. **The Polling Phase**

   During this phase the voters vote with their anonymous passes obtained in phase one. Each voter can only vote once. This phase can be divided into two sub-phases. Namely the voter authentication phase and the poll-casting phase. The polling phase can also be regarded as the polling period. The polling phase and the registration phase can run simultaneously. After the Administrator accepted a valid vote from the voter, the public counter will be increased by one.

3. **Vote Opening Phase**

   This phase begins after the polling phase. The result is computed and published.

In our system, the voters can vote in any locations with any devices that can be connected to the network. However, they are required to vote in a voting station for absolute uncoercibility.

7.3 Two Cryptographic Protocols

This section explains two cryptographic protocols specially designed for our voting scheme. In section 3.1 we describe an authentication protocol which is innovated by the E-Cash protocol [19]. In section 3.2 we describe a modified commitment protocol.

7.3.1 Protocol One - The Anonymous Authentication Protocol

The anonymous authentication protocol is designed for supporting the non-transferable anonymous voting pass of our system. We have modified the *check withdrawal transaction protocol* in [19] so that the following desirable properties can be achieved:

1. Alice can prove the knowledge of her secret key anonymously.
2. Anyone who passes the random challenge must possess the knowledge of Alice's secret key. This is to ensure that Alice cannot transfer the pass to other parties or she risks the disclosure of her private key.
3. Unlike the original protocol in [19], the secret information is not released to the Administrator during the registration phase.

We are going to describe how Alice obtains a non-transferable pass from Bob. Let Alice's private key be u and the corresponding public key be v, where $u \times v \equiv 1 \pmod{\phi(q)}$ is the published modulus [9]. We also define the operation $h(x)$ as

$$h(x) = \alpha^x \pmod{n}$$

where α is a published element in $GF(q)$. $h(x)$ is a one way function. It is computationally difficult to calculate x from $h(x)$ and α [45].

The description of Protocol One is as follows:

1. Alice first generates k candidates (called T_i) in the following way. She generates the random integers a_i, b_i, c_i and r_i for i = 1 to k. Then she computes:

$$x_i = g(a_i + u, b_i)$$

$$y_i = g(-a_i, c_i)$$

$$\text{and } T_i = {r_i}^e \cdot f(x_i, y_i) \pmod{n}$$

Where (e, n) is the public key of Bob. f, g are some two-argument collision-free functions. $a_i + (-a_i) \equiv 0 \pmod{F(q)}$, where (v, q) is the public key of Alice.

2. Alice presents the k T_i to Bob.

3. Bob randomly splits the integers 1,...,k into two ordered sets, S and S', each consists of $k/2$ elements. He then sends S to Alice.

4. Alice sends the values of $h(-a_i)$ and r_i for all $i \in S$ to Bob.

5. Bob generates a $k/2$-bit binary random challenge vector $\mathbf{z} = (z_0, z_1, ..., z_{k/2})$ and sends it to Alice. The i^{th} bit in z corresponds to the i^{th} element in S.

6. Alice responds according to the following rule:
   - when $z_i = 0$, Alice sends Bob $a_j + u, b_j$ and y_j;
   - when $z_i = 1$, Alice sends Bob $-a_j, c_j$ and x_j.
   Where j is the i^{th} element in S.

7. Bob verifies accordingly:
   - when $z_i = 0$, Bob checks if $(h(a_j + u) \cdot h(-a_j))^v \equiv \alpha \pmod{q}$. He then checks if the corresponding T_i are valid;
   - when $z_i = 1$, Bob checks if the $h(-a_i)$ presented by Alice previously can be derived from $-a_i$. He then checks if the corresponding T_i are valid.

8. If T_i is valid for all $i \in S$, Bob signs on the other $k/2$ unopened T_i with his private key d and returns to Alice:

$$\prod_{i \in S'} T_i^d \pmod{n}$$

9. Alice removes the blinding factors and form

$$\prod_{i \in S'} f(x_i, y_i)^d \pmod{n}$$

Which is Bob's signature on the pass

$$\Psi = \prod_{i \in S'} f(x_i, y_i) \pmod{n}.$$

Notice that steps (4) to (7) are necessary in order to ensure that Alice's private key, u, is included in the pass Ψ.

When Alice shows her pass to Bob, the following steps will be taken:

10. Alice presents the anonymous pass Ψ to Bob.

11. Bob checks if the signature on Ψ is valid.

12. Bob generates a $k/2$-bit binary random challenge vector $\mathbf{z} = (z_0, z_1, ..., z_{k/2})$ and sends it to Alice. The i^{th} bit in z corresponds to the i^{th} element in S'.

13. Alice responds according to the following rule:

- when $z_i = 0$, Alice sends Bob $a_j + u, b_j$ and y_j;
- when $z_i = 1$, Alice sends Bob $-a_j, c_j$ and x_j.

Where j is the i^{th} element in S'.

14. Bob checks if Ψ can be derived from the partial openings. If so, authentication will be granted.

Notice that Alice remains anonymous to Bob because both Ψ and the partial openings do not contain plain information on Alice's identity. Furthermore, one must know the values of $a_i + u, -a_i, b_i$ and c_i for all i in S' in order to succeed in the challenge on Ψ. This means that the proving party is able to access Alice's private key u since

$$a_i + u + (-a_i) \equiv u \pmod{F(q)} \text{ for any } i \text{ in } S'.$$

Therefore if Alice sells her pass to a third party, she is releasing her private key u as well.

7.3.2 Protocol Two - Anonymous Commitment

In our system, the voter signs and commits on the vote anonymously. Neither the voter can deny a committed vote nor the administrator can change an accepted vote. This is achieved by the anonymous commitment protocol. The identity of a voter is only revealed to the TTP when disputes occur.

We are going to illustrate how Alice commits on a message m using her private-public key pair (u, v) anonymously.

Let $H(x)$ be a multiplicative homomorphic one-way hash function as described in [46], and we define the two-argument hash function $H(x, y)$ as $H(x \cdot y)$. Due the multiplicative homomorphic property,

$$H(x, y) = H(x \cdot y) = H(x) \cdot H(y).$$

To commit on m, Alice submit the followings to Bob:

$$(m||H(m)||H(Sign_u(m))||H(m, Sign_u(m))$$

where $Sign_u(m)$ denotes Alice's signature on m using her private key u. Bob accepts the commitment if $H(m) \cdot H(Sign_u(m)) = H(m, Signu(m))$. Notice that Alice remains anonymous.

We give a brief security analysis of protocol two here.

1. Alice cannot deny the committed message m because Bob cannot generate $H(Sign_u(m))$.

2. Bob cannot change m because he cannot generate a valid $H(Sign_u(m'))$ on m'.

3. Alice cannot cheat by submitting the followings

   $$(m||H(m)||H(Sign_u(m'))||H(m, Sign_u(m'))$$

   to Bob but claiming m is modified from m' by Bob. This is because the TTP will ask Alice to sign on m', which she claims the original message is. It is easy to find out that $H(Sign_u(m')) \cdot H(m') \neq H(m, Sign_u(m'))$.

Both Protocol One and Protocol Two will be employed in our voting protocol. This will be introduced in the next section.

7.4 The Electronic Voting Protocol

The main innovation in our protocol is the employment of a non-transferable anonymous voting pass, which is innovated by the E-Cash scheme [19]. In addition, the modified commitment scheme eliminates the submission of a vote-opening key by the voters. In this section we describe the proposed E-Voting

protocol phase by phase.

7.4.1 The Registration Phase

Suppose Alice is a valid voter in the election. Before she polls, she needs to register for the election and obtains an anonymous voting pass from the Registrar. The Registrar holds a list of names of the valid voters. Step (1) to (9) of Protocol One is run, with Bob being replaced by the Registrar. The resulting pass ψ is the voting pass.

7.4.2 The Polling Phase

Two successive sub-phases are involved in the polling phase. Namely the voter authentication and vote-casting processes. The Administrator is responsible for the execution of this phase.

Voter Authentication

The authentication process is done by step (10) to (14) in Protocol One. With Bob being replaced by the Administrator. Authentication is granted upon the successful verification on an unused ψ.

Poll-Casting

The poll-casting phase is run immediately after the voter authentication phase. Protocol Two is run, with Bob being replaced by the Administrator. Also, the message m is replaced by the ballot β. Alice submits the following encrypted token

$$Encrypt_c(\beta||H(\beta)||H(Sign_u(\beta))||H(\beta, Sign_u(\beta))$$

to the Administrator, where $Sign_u(\beta)$ denotes the signature of β with u, the private key of Alice. The entire token is encrypted with c, the asymmetric encryption key of the Collector. The Collector is the digital analogy of submitting the ballot into the voting box.

After accepting the ballot from the valid voter, the public counter will be increased by one. A reference number is also recorded and saved at the Administrator.

The voter authentication sub-phases and the vote-casting sub-phase are two non-separable processes. If interruption occurs, the first sub-phase should be run again. This is necessary in order to ensure that the ballot is submitted by the valid pass holder.

7.4.3 Vote-Opening Phase

After the polling period, the Collector sends the decryption key c' to the Administrator. The Administrator decrypts the tokens collected in the vote-casting phase, it also verifies and counts the votes.

The final result is calculated. A result list, which is shown in Table 7.1 is published.

Given n is the total number of ballots collected, the first column shows the reference number of a particular ballot; and the second column lists the content of the ballot in concern. Notice that n should equal to the final reading on the counter. It is easy to check if the Administrator has varied any ballots. A particular voter can also check if her vote has been counted correctly. This will

	Received Ballots
1	$(\beta_1 \| H(\beta_1) \| H(Sign_{u1}(\beta_1)) \| H(\beta_1, Sign_{u1}(\beta_1)))$
2	$(\beta_2 \| H(\beta_2) \| H(Sign_{u2}(\beta_2)) \| H(\beta_2, Sign_{u2}(\beta_2)))$
3	$(\beta_3 \| H(\beta_3) \| H(Sign_{u3}(\beta_3)) \| H(\beta_3, Sign_{u3}(\beta_3)))$
.	.
.	.
.	.
n	$(\beta_n \| H(\beta_n) \| H(Sign_{un}(\beta_n)) \| H(\beta_n, Sign_{un}(\beta_n)))$

Table 7.1: Comparison of Computational Speed of Hash Functions Evaluation against other Operations

be discussed in next section.

7.5 Security Analysis

Our electronic voting protocol satisfies the basic security requirements listed in some of the previous electronic-voting papers, such as those listed in [39] and [47]. In additional, our protocol is a receipt-free scheme, in which the voters cannot sell the votes to the others using the receipts. Also, neither the voter can deny a committed vote nor the administrator can change an accepted vote.

7.5.1 Basic Security Requirements

According to [39], the basic security requirements in an election system include completeness, soundness, privacy, unreusability, eligibility, fairness and verifiability. In this subsection, we analyze the security of our systems in terms of these parameters.

Completeness

In [39], completeness is interpreted as *all valid votes are counted correctly.* Our system satisfies the completeness requirement because the Administrator cannot change or drop an accepted vote. The final number shown on the public counter indicates the number of ballots collected. Any drop of votes can be detected by the incorrectness of the counter. Moreover, our system prevents the Administrator from miscounting the votes because it is possible to determine if the voting result is correct from the published table. Also, the voters are allowed to check their voting record at a trusted third party (TTP) after the vote-opening phase.

When a voter wants to check her voting record, she presents ψ together with the request to the TTP. Upon receiving the request, the TTP verifies ψ using step (9) to (13) in protocol two, with Alice and Bob being replaced by the voter and the TTP respectively. If the verification is successful, it asks the Administrator for the reference number of the corresponding vote. This reference number will not be signed by the TTP and it is passed to the voter. She can check against the published list and see if her vote has been counted correctly. The voter remains anonymous.

As illustrated in Protocol Two, any change of vote by the Administrator can be detected. Further actions will be taken if disputes occur.

Soundness

Soundness refers to the inability of a dishonest voter to disrupt the voting. In our system, the voter cannot deny a committed vote, nor she can frame up the Administrator for changing his vote. This has been explained in section 3.2 and we do not repeat here.

Privacy

Privacy ensures a voter's vote being kept secret such that no one other than the voter knows her choice. Since the voting passes in our system are anonymous, voter privacy is provided. The voter's identity is only revealed to a trusted third party when dispute occurs.

Unreusability

In our system, no one is able to vote twice. Any reuse of the voting pass is disallowed. The Administrator makes sure that every pass is used only once during the voter authentication phase.

Eligibility

In an eligibility-supported voting system, anyone who is not allowed to vote cannot vote. In our system, the voting right of a voter is validated during the registration phase. Therefore the voting pass is only issued to the eligible voters. Multiple Registrars can be employed [48] in order to prevent a cheating Registrar from issuing voting passes to ineligible voters.

Moreover, the voting passes in our system are non-transferable. No ineligible voters can vote by buying a voting pass.

Fairness

In a fair voting system, nothing can affect the voting [39]. Our system is a fair system. Firstly, the Administrator cannot change a committed ballot. Secondly, a voter cannot sell her vote using the receipt because our system is receipt-free.

Thirdly, the voting passes are non-transferable so that the unauthorized parties cannot vote. Lastly, the final result is published and can be verified.

Verifiability

According to [39], the verifiability is interpreted as whether there are any parties who can falsify the result of voting. In our system, the overall election results as well as the ballots are published. In addition, the voters are allowed to check their votes. Therefore our system supports the verifiability requirement.

7.5.2 Receipt-freeness

In some early electronic voting proposal such as [19], [39], [49] and [50], the voters are able to show to a third party how the vote is cast. Such systems are not receipt-free and they are against the fairness requirement of a voting system.

A definition of receipt-freeness has been given in [40]. According to this definition, an electronic voting system is receipt-free if the voter V_i can cast a vote $v_i^* \neq v_i$ and is accepted by the coercer, who wants to interfere the voting decision of V_i; while v_i is the favorite vote of the coercer. Our system is a receipt-free system because the voter has no evidence to prove how her vote has been cast. Unlike the scheme proposed in [39], the Administrator issues no signed document on how the vote has been received. Also, the bulletin board lists only the received ballots without the voter information. Therefore the voters cannot show how they vote in the election.

Even if the voter checks her vote at the TTP after the election, no receipt will be generated and she still cannot prove her vote to the coercer. Notice that the checker must possess the corresponding private key resides in Y as explained

in section 3.1. Therefore only the voter in concern can check her voting record and she cannot prove her vote to a third party directly.

7.5.3 Non-transferability of Voting Right

We believe that the voting right must not be transferable in an eligible voting system. This requirement, however, is violated in a number of electronic voting proposals such as [31], [42], [43] and [44].

We have proposed the non-transferable voting pass in our paper. The voting pass contains the information of the private key of the holder. Our algorithm has been carefully designed so that the pass issuer can ensure that the private key is included in the pass, while no information about the private key is revealed. As illustrated in section 3.1, one must know the values of $a_i + u$, $-a_i$, b_i, and c_i for all i in S' in order to succeed in the challenge on the voting pass. This means that the proving party must be able to access Alice's private key u since

$$a_i + u + (-a_i) \equiv u \pmod{F(n')} \text{ for any } i \text{ in } S'.$$

Therefore if Alice sells her pass to a third party, she is releasing her private key u as well. This explains why the voting pass is non-transferable.

7.6 Conclusion

In this chapter, we have pointed out the requirement of non-transferability of voting right. Two cryptographic protocols, the anonymous authentication protocol and the anonymous commitment protocol, are presented. We have proposed a new electronic voting system in which the voters vote with the non-transferable

voting passes, such voting passes are achieved by the anonymous authentication protocol. In addition, our system is receipt-free and it satisfies the basic security requirements include completeness, soundness, privacy, unreusability, eligibility, fairness and verifiability.

Chapter 8

Anonymous Internet Access

8.1 Introduction

The rapid development of the Internet has made a revolution over our daily life. Information is abundantly and easily accessible to everyone who has a connection to the world wide information superhighway. Internet applications like electronic commerce, electronic messaging (e.g. E-mails) as well as the World Wide Web provide great convenience to the modern society and eventually transform commerce, education, provision of government services and almost every other aspect of modern life.

However, the privacy issues accompanying these innovations cannot be neglected. The potential interception or misuse of personal data collected from the provision of Internet services is a threat to user privacy. Moreover, the rapid development of data mining technologies makes this threat even more severe. This results in the calling of anonymous Internet access.

We believe that anonymity should be provided conditionally. In the later

part of this chapter, we will propose a cryptographic solution that can achieve conditional anonymous Internet connections, which is developed based on the Electronic Cash Protocol introduced in [19]. In our protocol, user anonymity is maintained so long as the user does not misbehave. In this way, user privacy is protected while anonymity is not abused.

8.2 Privacy Issues of Internet Access Services

Internet services providers can learn much about their customers, as all information that will pass to an Internet user must first pass through the proxies reside in their servers. Although encryption techniques such as SSL are used so that third parties cannot interpret the contents being transmitted, the ISP can still determine what web sites or even which article a particular user has visited. This is because every Internet object request originated from the users is logged in the ISP's proxy cache. This is referred as the 'clicktrails' data collection. Collecting and analyzing the 'clicktrails' data can derive much information about a person.

8.2.1 Present Privacy Laws and Policies

The collection of personal data is unavoidable in many occasions (e.g. opening a bank account), Some existing privacy ordinances such as [51] allows service providers to collect user's private information for the purpose intended, but prevents them from changing the usage of such data. For the case of Internet services provision, at present, an ISP has right to collect 'clicktrails' data and hold log files of user Internet usage for the purposes of system maintenance and

troubleshooting. Individual ISP also has own policy on privacy, however the standard diverges and user privacy is sometimes not properly protected.

8.2.2 Present Anonymous Internet Services Solutions

Several anonymous Internet services technologies have been developed. They aim at hiding the user identity from the remote sites. For example, the anonymous web servers such as [52] fetches the requested objects on behalf of the users, so that the remote host receives the request apparently originated from the server. There are also technologies such as the Onion Routing [53] providing anonymous connection in which routing information is hidden.

8.2.3 Conditional Anonymous Internet Access Services

In order to prevent the abuse of the data in log files, users should remain anonymous to the ISP during Internet object requests. This can be achieved by using cryptographic methods. In Section three of this paper, we will present a conditional anonymous Internet access protocol that have the following features:

- User is anonymous to the ISP during Internet access.
- The ISP has no way to relate a requested object to its requester even if it is fetched via the ISP's proxy.
- The anonymity is conditional, the user identity is revealed when any misbehavior is detected.
- This protocol is transparent to other applications; and it is interoperable among different ISP's.

Here we assume the employment of caller-ID blocking so that the ISP cannot trace the user identity from the phone number.

8.3 The Protocol

We developed our protocol motivated by the E-Cash protocol proposed in [19]. In our system, the user login with a pass

$$(x, f(x)^{1/3} \pmod{n})$$

in which the first term is the pseudonym of the user and the second term is the ISP's public key signature [9]. Here n is a published composite and only the ISP knows its factors, $f(.)$ is a one-way function known by both the user and the ISP and x is the pseudonym chosen by the user. We firstly introduce a simpler version of the protocol, in which anonymous Internet connection without user identity recovery is achieved. In later section, we will discuss how the protocol is modified so that the user's identity can be revealed in case of any misbehavior occurrence.

8.3.1 ISP issues a new pass to Alice using blind signature [1] scheme

To open a new account, Alice generates x and r, where x is a pseudonym with which she will use to access the Internet services and r is a blinding factor. These numbers only known by Alice. She presents the following token:

$$T = r^3 \cdot f(x) \pmod{n}$$

to the ISP.

Upon receiving T and having verified on Alice's identity, the ISP signs on T by calculating the third root of T modulo n and returns $T^{1/3} \pmod{n}$, i.e. $r \cdot f(x)^{1/3} \pmod{n}$, to Alice. It is assumed that only the server has the knowledge to compute the third root modulo n [9]. Alice then extracts $f(x)^{1/3} \pmod{n}$ from the returned token by dividing $T^{1/3} \pmod{n}$ with r and form her pass:

$$pass = (x, f(x)^{1/3} \pmod{n}).$$

This pass is saved at Alice's side. She logins with this anonymous pass instead of her login name from now on. The server has no way to relate Alice to her pseudonym x because it cannot see the value of x when it signs on T.

8.3.2 Account Operations

An account for x is created, where x is the pseudonym of Alice. When Alice login, she presents the pass *pass* instead of using her own username and password. Authentication is done by verifying the value of $f(x)^{1/3} \pmod{n}$. from pass. All other account operations are similar to the existing system. Since the server has no knowledge on the identity of x, anonymous Internet service provision is achieved.

8.4 Modified Version with Key Escrow on User Identity

In Section 3 we have presented the simpler version of our protocol. In this section, we modify on it so that the following desirable additional properties can be achieved:

- Alice is the only legitimate user of the pass.
- Alice's identity will be revealed when necessary.

The modification is based on the double spending prevention solution presented in [19]. The later property enables identity revocation in critical situations. In the modified version of the protocol, a trusted third party (TTP) is involved. And a valid pass has the following format:

$$(pseudonym, \{pseudonym\}_{ISPsign}, \{pseudonym\}_{TTPsign}).$$

Here we make the same assumption that only the ISP has the knowledge to compute the third root modulo n.

8.4.1 Getting a new pass

Let f and g be some two-argument collision-free functions as described in [19]. And let u be a unique identifying number of Alice (e.g. the account number). Instead of producing a single blinding factor r as in the previous section, four independent sets of random numbers each consists of k elements, $\mathbf{a}$, $\mathbf{c}$, $\mathbf{d}$ and $\mathbf{r}$ are generated.

In order to obtain the blind signature from the ISP, Alice forms and sends k T_i's in the following manner:

$$T_i = r_i^3 \cdot f(x_i, y_i) \pmod{n}$$

where

$$i = 1 \text{ to } k$$
$$x_i = g(a_i, c_i)$$

and

$$y_i = g\ (a_i \text{ XOR } u,\ d_i).$$

Notice that at this stage, The ISP knows Alice's identity, u. In order to verify the T_i's presented by Alice, the ISP undergoes the following steps:

1. It chooses randomly a set of $k/2$ integers, $R = \{i_j\}$, where $1 \leq i_j \leq k$ and $1 \leq j \leq k/2$.

2. It asks Alice to show the values of r_i, a_i, c_i and d_i for every i in R.

3. It compares the $k/2$ presented T_i's and see if it can be derived from these r_i, a_i, c_i, d_i and u.

After that, the ISP gives Alice

$$\prod_{i \notin R} T_i^{1/3} \pmod{n}$$

And Alice can easily extract the following component:

$$\prod_{i \notin R} f(x_i, y_i)^{1/3} \pmod{n}$$

which is correspond to the ISP's blind signature on the pseudonym,

$$p = \prod_{i \notin R} f(x_i, y_i) \pmod{n}.$$

Notice that the ISP has no way to relate u to p because it cannot see x_i and y_i for $i \notin R$.

Alice also needs to get the signature from the TTP. Before signing on p, the TTP verifies the validity of p and writes part of the information about Alice's identity into the database. Notice that Alice is anonymous to the TTP and the information obtained by the TTP is not enough for it to compute Alice's identity.

To perform this task, the same set of k T_i's are presented to the TTP. The TTP then performs the following procedures:

1. It asks Alice to give the values of r_i, x_i, (a_i XOR u), and d_i for every i.
2. It verifies if the corresponding T_i's can be derived from the presented values.
3. If the verification succeeds, the TTP stores the values (a_i XOR u) along with p into the database.

The TTP then verifies and signs on the pseudonym p:

1. Base on the values collected from the above stage, it verifies if p is indeed constructed from the T_i's where $R' = \{i \in Z : i \notin R, 1 \leq i_j \leq k\}$.

2. It signs on the pseudonym using normal public-key signature schemes.

Upon receiving the TTP's signature, Alice can then form the pass:

$$(pseudonym, \{pseudonym\}_{ISPsign}, \{pseudonym\}_{TTPsign}).$$

In the process of pass verification just mentioned above, the cryptographic method, zero-knowledge proof is employed. It enables one to prove his/her identity to the other party without revealing the identity. More details can be found in [54].

8.4.2 Account operations

Account operations can be performed as usual. However, there are some differences in pass verification.

The verification of a pass includes three procedures, namely the verifications of the ISP signatures and that of the TTP signatures, and also the process to ensure Alice (who is anonymous to the ISP) is indeed a valid holder of the pass.

The first two processes can be performed directly using the public-key signature verification schemes. While the last part is done by the following:

1. The ISP generate a random binary vector $Z = (z_1, z_2, ..., z_{k/2})$ where the element z_i correspond to the i^{th} number in the set R'.

2. Alice responds according to the following rule:

   - when $z_i = 1$, Alice send the ISP a_i, c_i, and y_i.
   - when $z_i = 0$, Alice send the ISP x_i, and y_i.

3. From the received values, the ISP can check if the corresponding values satisfies the pass.

8.4.3 Identity revocation

In this protocol, a user remains anonymous so long as he/she does not misbehave. In this way, a limited anonymity is provided so that user privacy cannot be abused. This is done by the cryptographic method of secret splitting; in which a piece of secret is divided among two or more parties and each party alone does not have knowledge about the secret [55].

When misbehavior of the holder of p is detected or in case of any appeals, the court asks the ISP to presents the pass p along with a_i for $i \in R'$; which it obtains during pass verification process. The court also gathers the corresponding (a_i XOR u) for $j \in S$ where $S = R \cup R'$ from the TTP. Notice that $S \cap R' \neq \{\emptyset\}$ and let e be an element in $S \cap R'$. The user identity u is revealed as:

$$u = a_e \text{ XOR } (a_e \text{ XOR } u).$$

8.5 Security Analysis

This section analyzes on the strength of our protocol in resistance of different potential threats.

8.5.1 Anonymity

Alice remains anonymous to the ISP. This is because during the stage of pass issuing, Alice prepares k candidates of T_i's and the ISP only random challenges on $k/2$ of them. The other $k/2$ values which are used to form the pass are never seen by the ISP. During the authentication process, the ISP only random challenges on the $k/2$ unseen values. Therefore the ISP cannot relate the identity

of Alice to the pass that she possesses.

8.5.2 Masquerade

During the pass issuing stage, Alice is not anonymous and the ISP should make sure Alice's identity before signing on the pass. This can be done by employing a digital certificate scheme; in which one's identity is proved by a recognized digital certificate. In this way, the identity of the pass receiver can be ensured.

8.5.3 Alice cheats

Since the ISP views only half of the k candidates of r_i, a_i, c_i and d_i, where $i \in R$. Therefore in the pass issuing process Alice may have chance to cheat. She does this by not using a valid u in the calculation of those $k/2$ T_i's which are not viewed by the ISP. However her chance of successful cheating decreases exponentially with the value of k. For example, when k equals 16, the chance for the ISP choosing none of the Alice's cheated T_i's is $1/2^8 = 0.0039$. When k increases to 32 the chance further decreases to $1/2^{16} = 1.526 \times 10^{-5}$.

8.5.4 Stolen pass

Suppose Alice's pass is stolen by Carol during the pass issuing stage, this will not bring any lost to Alice because Carol does not know the secrets that Alice is holding. That is, Carol does not know a_i, c_i and y_i for every $i \in R'$ which involve in the random challenge process during future logins. Therefore Carol cannot use the pass.

Suppose Carol steals the pass at later stages so that she also steals the

numbers a_i, c_i and y_i for some $i \in R'$. In this case, however, she cannot use the pass until she obtains a_i, c_i and y_i for every $i \in R'$. This is because different elements in **a**, **c** and **y** are challenged randomly each time so Carol can only obtain some of them each time. When Carol has waited long enough to collect a_i, c_i and y_i for every $i \in R'$, Alice may has already changed her pass.

8.6 Efficiency

Compare with the non-anonymous Internet access scheme, our secure anonymous Internet access protocol may require more computational power. In this section we analyze on the computational effort involved.

8.6.1 Random number generation

During the initial stage where the pass is issued, a total number of $4k$ random integers need to be generated. Where k is suitable and large enough to prevent cheat from any potential parties, as explained in Section 5.3. For example, if k = 32, then 128 random numbers are generated.

The number-of-bit of these random numbers are arbitrary. For example, when 32-bit binary numbers are used, the possible variation for the values of r_i, a_i, c_i, d_i equals 2^{32} = 4294967296. When 64-bit binary number is used instead, the number of possible variations of these values is increased to 2^{64} = 18446744073709551616. The higher the number-of-bit, the more secure but slower of the system is.

8.6.2 Signing on the pass

Blind signature by the ISP

When the ISP makes a blind signature on Alice's pass, it needs to verify the pass using cut-and-choose method. This involves verification on the $k/2$ presented T_i's; and each of the verification involves 3 hashes. The ISP also need to verify Alice's identity and this involves one public-key certificate verification. Therefore the ISP needs to undergo $3k/2$ hashes, one certificate verification and 1 public-key signature when it signs on a pass.

Signature by the TTP

For the TTP, two procedures are involved in the signing process. Firstly it verifies the x_i, (a_i XOR u), and d_i for every i. This involves $2k$ hashes. Secondly checks if the $k/2$ T_i's involves in the pass are valid. This requires another k hashes. Therefore the TTP needs to undergo $3k$ hashes and 1 public-key signature when it signs on a pass.

8.6.3 Pass validation

When a user login, the ISP undergoes random challenge and checks if the user is a legitimate holder of the pass. This involves a generation of a $k/2$-bit random binary number and $2z$ hashes, where z is the number of 1's in the binary number and $z \in k/2$.

8.6.4 Identity recovery

When the user misbehaves, his/her identity is going to be revealed. This simply requires one searching and one XOR calculation.

To conclude, most operations undergone in our protocol are hashes, and they are light in terms of computational power [56].

8.7 Conclusion

In this chapter we have pointed out the privacy problems involved in Internet access. We also proposed a cryptographic solution to the problem; which is motivated by the electronic cash protocols. Our protocol supports anonymous user login to a proxy server so that the Internet usage habit of a user cannot be traced and analyzed. However the user cannot abuse his/her anonymity because our protocol enable a misbehaved user's identity to be revealed. This is achieved by a key escrow method in which a trusted third party keeps half of the secret about the user's identity.

In addition, our protocol resides on the application layer and does not require changes in other layers during implementation. With suitable legislation, user privacy of Internet access can be properly protected.

Chapter 9

Conclusion

In this thesis, we have made a thorough study on electronic money. We started from the history of money. The properties and the requirements of money are also described. Because of the change in the way people conducting business, from traditional commerce to digital commerce, a new form of medium of transaction is required. In Chapter 4 and chapter 5, two main categories of electronic money: Electronic Cash and Micropayments are discussed. Comparisons between *macropayment* (electronic cash) systems and *micropayment* systems are made.

However, the contribution of electronic money is not only limited to providing a new form of medium of transaction in the Digital Age. Applications such as electronic voting and Internet privacy provision are derived from the electronic cash protocols. In this thesis, three innovative cryptographic applications that derived from electronic cash protocols are proposed. Namely the *Anonymous Micropayment Tickets*, *Non-transferable Electronic Voting Passes* and the *Anonymous Internet Access Service*.

In the Anonymous Micropayment Tickets protocol, a new concept: *Micropayment in form of a ticket* is proposed. In traditional micropayment protocols, security requirements have to be loosed because of their efficiency and low processing cost. In most case, anonymity is not provided. In the micropayment ticket protocol, a ticket can be used for more then once so that the overheads can be used repeatedly thus saves the computational cost. Elliptic curve digital signature is used for double spending prevention. This protocol is very suitable for implementation on the devices that contain a user certificate (can be anonymous or non-anonymous, thus may or may not retain its anonymity).

The Non-transferable Electronic Voting Passes, which is proposed in chapter 7, enable a more secure receipt-free electronic voting system. In the proposed system, a voter votes by registering anonymously and obtaining a voting pass. By carefully designed cryptographic algorithms, the voter's private key is embedded inside the pass so that only the true owner of the voting pass can use the pass. This is a new feature added to the cryptography-based receipt-free electronic voting systems. In this chapter, we have also proposed the anonymous commitment protocol, which eliminates the submission of a vote opening key by the voters after the polling period. The Non-transferable Electronic Voting Passes has been implemented on the Java platform and a test system is being used in the Department of Information Engineering, the Chinese University of Hong Kong.

In chapter 8, the electronic cash protocol is modified so that it supports an application which is quite unrelated to electronic cash. The Anonymous Internet Access protocol, which is derived from the protocol proposed in [19] provides

Internet user privacy by establishing conditional anonymous accounts. The implementation of this protocol only require modifications in the application layer, therefore is very convenient for the existing systems. Moreover, this protocol can be applied to the anonymous service provision of other applications such as Internet auction and anonymous certificates.

To conclude, electronic cash is important in the Information Revolution. This is not only because it can support the ever-growing electronic commerce in terms of a medium of transaction, but also because of its potential contributions. This thesis only lists three of the possible variations. We also believe that upon further modifications and development, electronic cash will gain a majority share on the market.

Appendix

Papers Derived from this Thesis

1. Y. Chan. On Privacy Issues of Internet Access via Proxy Servers. *Proceedings of CQRE*, Lecture Notes in Computer Science, Springer-Verlag, 1999.

2. Rosanna Y. Chan, Jonathan C. Wong, Alex C. Chan. Anonymous Electronic Voting System with Non-Transferable Voting Passes. *Proceedings of SEC2000*, to appear.

Bibliography

[1] David Chaum. "Blind Signatures for Untraceable Payments," *Advances in Cryptology - Proceedings of CRYPTO'82*, 1982.

[2] "The Global Market Forecast for Internet Usage and Commerce." *IDC*, June 1999.

[3] "Mondex Electronic Cash." [web page], 2000. http://www.mondex.com/ [Accessed 17 April 2000]

[4] H. Feistel. "Cryptographic Coding for Data-Bank Privacy." RC 2827, IBM Research, 1970.

[5] National Bureau of Standard. "Federal Information Processing Standard (FIPS), Publication 46: The Data Encryption Standard." 1977.

[6] National Institute of Standards and Technology (NIST). "Federal Information Processing Standard (FIPS) Publication 46-1: Data Encryption Standard." 1988.

[7] American National Standards Institute (ANSI). "ANSI X9.17-1985: Financial Institution Key Management." 1985.

[8] Lai, X. "On Design and Security of Block Ciphers." *ETH Series in Information Processing*, Vol. 1, Konstanz, Hartung-Gorre Verlag, 1992.

[9] Rivest, R.L., A. Shamir, and L. Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems." *Communications of the ACM*, v. 21, n. 2, 1978.

[10] M. O. Rabin, R. DeMillo, D. Dobkin, A. Jones, and R. Lipton. "Digitalised Signatures." *Foundations of Secure Computation*, 155-168, Academic Press, 1978.

[11] N. Koblitz. "Elliptic Curve Cryptosystems." *Mathematics of Computation*, v.48, n. 177, pp.203-209, 1987.

[12] V. S. Miller. "Use of Elliptic Curves in Cryptography." *Advances in Cryptology - Proceedings of CRYPTO'85*, 1985.

[13] N. Koblitz. "Algebraic Aspects of Cryptography." Berlin, Springer-Verlag, 1999.

[14] Hendrickson, Robert A. "The Cashless Society." New York, Dodd, Mead and Company, 1972.

[15] Edward Bellamy. "Looking Backward." London, W. Foulsham and Co. Ltd., 1888.

[16] Honor Croome. "Introduction to Money." Methuen and Co. Ltd., 1962.

[17] E.Victor Morgan. "A History of Money," Penguin Books, 1965.

[18] Davies, Glyn. "A history of money from ancient times to the present day." Revised Edition. Cardiff: University of Wales Press, 1996.

[19] David Chaum, Amos Fiat, Moni Naor. "Untraceable Electronic Cash." *Advances in Cryptology - Proceedings of CRYPTO'88*, 1988.

[20] Tatsuaki Okamoto, Kazuo Ohta. "Universal Electronic Cash." *Advances in Cryptology - Proceedings of CRYPTO'91*, 1991.

[21] Donal O'Mahony, Michael Peirce, Hitesh Teware. "Electronic Payment Systems." Artech House, 1997.

[22] eCash Technologies. [web page], 2000. http://www.digicash.com/ [Accessed 17 April 2000]

[23] Boly, J. P., et al. "The ESPRIT Project CAFE." *Computer Security - ESORICS'94*, Third European Symposium on Research in Computer Security Proceedings., Lecture Notes in Computer Science, Vol. 875, 1994.

[24] Bosselaers, A., et al. "Functionality of the Basic Protocols." Technical Report, ESPRIT Project 7023 (CAFE), 1995.

[25] David Chaum. "Blind Signatures for Untraceable Payments." *Advances in Cryptology - Proceedings of CRYPTO'82*, 1982.

[26] Bos, J., and D. Chaum. "Smart Cash: A Practical Electronic Payment System." Technical Report, CWI-Report: CS49035, 1990.

[27] Medvinsky, G., and B. Clifford Neuman. "NetCash: A Desihn for Practical Electronic Currency on the Internet." *Proceedings of First ACM Conference on Computer and Communication Security*, 1993.

[28] CyberCash, Inc. [web page], 1999. http://www.cybercash.com/ [Accessed 17 April 2000]

[29] Rivest, R., and A. Shamir. "PayWord and MicroMint: Two Simple Micropayment Schemes." 1996, http://theory.lcs.mit.edu/ rivest/RivestShamir-mpay.ps

[30] Glassman, S., et al. "The Millicent Protocol for Inexpensive Electronic Commerce." *Proceedings 4th International World Wide Web Conference*, 1995.

[31] Michael J. Radwin. "An Untraceable, Universally Verifiable Voting Scheme." Seminar in Cryptology, 1995.

[32] Rosanna Y. Chan, et al.. "Anonymous E-Voting System with Non-Transrferable Voting Passes." to be appear, World Computer Congress, SEC2000, 2000.

[33] D. Chaum. "Achieving Electronic Privacy." Scientific American, 1992.

[34] Y. Chan. "On Privacy Issues of Internet Access Services via Proxy Servers." *Proceedings of CQRE'99*, 1999.

[35] M. Franklin, M. Reiter. "The Design and Implementation of a Secure Auction Service." *IEEE Transactions on Software Enginerring*, Vol. 22, No. 5, 1996.

[36] IEEE. P1363, Standard Specifications for Public Key Cryptography.

[37] Wenbo Mao. "A Simple Cash Payment Technique for the Internet." *Proceedings of ESORICE'96*, Springer-Verlag, 1996.

[38] David Chaum. "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms." *Communications of the ACM 24*, 2, 1981.

[39] Atsushi Fujioka, Tatsuaki Okamoto, Kazuo Ohta. "A Practical Secret Voting Scheme for Large Scale Election." *Advances in Cryptology - Proceedings of AUSCRYPT'92*, 1992.

[40] Josh Benaloh and Dwight Tuinstra. "Receipt-Free Secret-Ballot Elections (Extended Abstract)." *Proceedings of the 26th Annual ACM Symposium on the Theory of Computing*, 1994.

[41] Tatsuaki Okamoto. "Receipt-Free Electronic Voting Schemes for Large Scale Elections." *Proceedings of Security Protocols'97*, 1997.

[42] Yi Mu and Vijay Varadharajan. "Anonymous Secure E-Voting over a Network." *Proceedings of 14th Annual Computer Security Applications Conference*, 1998.

[43] Qi He and Zhongmin Su. "A New Practical Secure e-Voting Scheme." *Proceedings of SEC'98*, 1998.

[44] A. Riera, J. Borrell and J. Rifa. "An Uncoercible Verifiable Electronic Voting Protocol." *Proceedings of SEC'98*, 1998.

[45] K. S. McCurley. "The Discrete Logarithm Problem." *Cryptology and Computational Number Theory*, v.42 of Proceedings of Symposia in Applied Mathematics, 1990.

[46] Josh Cohen Benaloh. "Secret Sharing Homomorphisms: Keeping Shares of a Secret Secret (Extended Abstract)." *Proceedings of CRYPTO'86*, 1986.

[47] Ronald Cramer, Matthew Franklin, Berry Schoenmakers and Moti Yung. "Multi-Authority Secret-Ballot Elections with Linear Work." *Proceedings of EUROCRYPT'96*, 1996.

[48] Brandon William DuRette. "Multiple Administrators for Electronic Voting." MIT Thesis, 1999.

[49] J. Cohen and M. Fischer. "A Robust and Verifiable Cryptographically Secure Election Scheme." *Proceedings of the 26th IEEE Symposium on Foundations of Computer Science*, 1985.

[50] Josh Benaloh and Moti Yung. "Distributing the Power of a Government to Enhance the Privacy of Voters." *Proceedings of 5th ACM Symposium on Principles of Distributed Computing*, 1986.

[51] The Government of Hong Kong SAR of the People Republic of China. Personal Data (Privacy) Ordinance, Version date 20 Dec 1996, 1996.

[52] The Anonymizer. [web page], 1999. http://www.anonymizer.com/ [Accessed 6 Sept 1999]

[53] Reed M.G., Syverson P.F., Goldschlag D.M. "Proxies for anonymous routing." *Proceedings, 12th Annual Computer Security Applications Conference*, 1996.

[54] Bruce Schneier. *Applied Cryptography 2nd Edition*, 1996.

[55] H. Feistel. "Cryptographic Coding for Data-Bank Privacy." RC 2827, Yorktown Heights, NY. IBM Research, 1970.

[56] D. O'Mahony, M. Peirce, H. Tewari. *Electronic Payment Systems*, 1997.