E-COMMERCE AND ITS DERIVED APPLICATIONS: SMART CARD CERTIFICATE SYSTEM AND RECOVERABLE AND UNTRACEABLE ELECTRONIC CASH

BY

LIU KAI SUI

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF PHILOSOPHY

IN INFORMATION ENGINEERING

JULY 2001

Acknowledgement

I would like to thank my supervisor Prof. Victor Keh-Wei Wei for his guidance on my research process. It is Prof. Wei who introduced me to the world of cryptography a few years before and provided me the best environment for research. Prof. Wei has provided me the best equipment for my research, such as the LCD monitor, the notebook and the fast computers which help me a lot. This thesis would not have been possible without his valuable ideas, teaching and support.

Also need to thank my dearest classmates, especially those in the Information Integrity Lab (iilab). Sandy and Chun Man always play with me and talk with me. We also like to discuss many interesting things together, including both academic and non-academic. Although Siu Chun, my FYP partner, is not sitting the iilab now, yet he helps me to improve my paper. Anthony Chan who sits next to me always helps me to solve many computer related problems. I like to talk to Simon Chan, Brian and Simon Shum and find it very interesting. Besides my lab, other classmates also provide me a relax environment so that my university life can become more colourful.

Last but not least, I have to say thanks to my mother for giving me so much care and support. Finally, I have to thank our Lord, Jesus Christ, for hearing my prayer.

Abstract

In the 21st century, many parts of our daily life have already been digitized. The ways of performing commercial activities are also changing. We can now go to the web for on-line shopping. We can go to the website of a bank to do some financial transactions. We can go to the website of our government to pay for taxes and apply for services. We call these kinds of digital commercial activities "Electronic Commerce".

Among the field of e-commerce, security is a main concern. Cryptographic primitives are needed to build up a secure environment. In this thesis, we make a thorough study on these primitives such as public-key cryptography and digital certificate. A new public-key algorithm *elliptic curve cryptosystem* is also introduced. Besides, we also make details description of two media generally used in e-commerce, namely *smart card* and *electronic cash*.

Our contributions also focus on these two media. The first one we proposed is called *Smart Card Certificate System* and the second one is called *Recoverable and Untraceable Electronic Cash*.

摘要

踏入二十一世紀，數碼化趨勢已經滲透到我們生活的方方面面。商業活動的模式也逐漸發生變化：我們可以去"網上商店"進行"在線購物"；可以用"網上銀行"完成"電子轉帳"；甚至可以到政府部門的網站繳付各項稅款或者申請各種服務 .....這些形形色色的數碼商業活動即是我們現在所講的"電子商務"的內容。

在電子商務的領域中，網絡安全是備受矚目的一環，因此爲了建立安全的交易環境，密碼學的基本要素不可或缺。本論文對於諸如 RSA 公匙密碼學、數碼證書等密碼學的多種基本要素進行了深入的研究。本論文同時也介紹了一種全新的公匙密碼學算法——橢圓曲線密碼學系統。除此之外，本篇論文亦對在電子商務中廣泛應用的兩種媒體——智能卡與電子錢，做了比較深入的探討。

本篇論文也集中於這兩種媒體的研究和探討。對於智能卡系 統，本論文提出了「智能卡數碼證書系統」方案；對於後者，我們則提出了「可復原但不可追蹤的電子錢」的概念和解決方案。

Contents

Chapter 1

Introduction

In the past, when the deadline of a payment (such as electricity bill or government tax) comes, there are many people lining in the queue of the payment service. The waiting time varies from a few minutes to an hour. It really wastes a lot of time. Later there is auto-pay service such that the electricity company or the government credits the user's account at a fixed time. But some people think that it is not good enough because they are afraid the company may overcharge them. They still want to pay the bill by themselves, yet they do not want to waste any time in the waiting queue. Recently they can have a way to achieve their goal. Now there are many method of payment, such as using digital telephone, WAP (wireless application protocol, used in mobile phone for Internet access), WWW etc.

Digital! We are now living in the Digital world. Nowadays many parts of our daily life have converted into the digital domain. Mobile phone, personal computer, digital camera, digital video (DV) recorder, mp3 player etc... Even the ways of communication between people are digital – using the Internet. Email, world wide

web (WWW), wireless application protocol (WAP), ICQ etc. They are the ways that people use to communicate with each other and obtain the newest information. Commercial activities are also digitized. We call this kind of activity Electronic Commerce.

Let us look into the new model for organizational e-commerce. There are three different viewpoints [33]:

1. The new organization, which was born on the Internet in the e-commerce market-space.
2. Established organization traditionally in an offline market-space now move to the Internet.
3. Organizations that are coming together in a new format – the e-consortia partnerships are made through the "virtual structure" of an online organization.

Figure 1.1 Established, online, and consortium organizations in the market-space.

In Figure 1.1 [33], we can clearly see that the e-commerce market-space is expanding. To become successfully in the e-commerce market-space, technology is a key factor. Among all the technology, security is the most important part of it. Without security, people will have no trust in the electronic systems and all the business will fail.

1.1 Security and E-commerce

In the paper cash commercial model, security is already a necessary factor people concerned. The security of each bank is higher than other shops. Extra cost or manpower is needed when transferring the money from one place to another. It is the same in the electronic domain. In e-commerce, we talk about data security instead of physical security.

Try to imagine that if the following cases happen:

1. The web server of your company is hacked by someone, and all the information of your customer is stolen.
2. The customer's credit card number is known by someone else during the transmission from his own computer to the web server.
3. A customer uses the e-banking service of a bank. Unfortunately, the password he typed during the online transaction of account is captured by someone.

The result is very serious. More importantly, customers will have no confidence in doing e-business or getting e-service because they think that it is insecure and they may have a high risk in doing that. Therefore, we can see that the foundation of e-commerce is security. Everything is built on top of it. If the foundation is not good enough, the whole e-commerce model will collapse.

Art Coviello, CEO of RSA Security, believe that *"companies are realizing PKI can enable key business applications"*. He also has a positive prediction over the IT security industries despite the economy slowdown, *"of all the categories in IT spending, security is the one last category that will likely be one of the last places to be cut, if cut at all"* [34]. Furthermore, according to [34], *"for the first time, security interests and business interests are aligned"*. We can find many real time example, such as Oracle Mobile is working with Sonera Corp subsidiary, SmartTrust, to add wireless PKI functions to the wireless version of Oracle's 9i database [34]. Certicom [35], a company selling Elliptic Curve Cryptosystem product, has already had alliance with Motorola [36] to provide the market with advanced levels of security for trusted wireless transactions. This is the "E-consortium" that we have shown in Figure 1.1

1.2 E-commerce: More than Commercial Activities

While e-commerce plays an important role in the society, its innovations are not only limited to commercial activities. B2B (business-to-business) and B2C (business-to-customer) are the most common models of e-commerce. Yet there are also other kinds of model. B2G (business-to-government), G2B (government-to-business) or even G2C (government-to-customer or citizen) are those newly developed models.

B2G can be further subdivided into horizontal market and vertical market [33]. For the horizontal market, Grainger [37], a company in the United State, has created its market-space across the U.S. Federal and State/Local government agencies. For the vertical B2G portal, they provide a specialized service to fill a special need of the government, such as processing of passport or fingerprints for the immigration department.

G2B and G2C can be discussed together because their properties are very similar. When the government provides an electronic service to fulfill the requests for information from business, it can be regarded as G2B. If the target of the service is citizen, it can be regarded as G2C.

Recently, the Hong Kong SAR Government also focuses on the G2C systems. We can now go to the website of the HKSAR Government [38] to download application forms or documentation, to apply for a driving or car license, to report criminal activities, to get the most updated news or report from the government, to apply for some government jobs etc. The ESDLife (Electronic Service Delivery Life) is a new service provided by the HKSAR Government to enable citizen to do different kinds of electronic services. It aims to deliver high quality public services to the community in an innovative manner; to improve the efficiency and reduce the cost of delivery of public services; and to foster the development of electronic commerce in the HKSAR [40]. How about in the U.S. side? On 7 March 2000, the first official Internet voting was conducted in Arizona! Citizen are no longer need to go to the voting center which maybe very far away from their home. They can use any computer to execute their citizen right – to vote for their representatives.

Yet no matter it is B2B, B2C, B2G, G2B or G2C, security is also the main concern. Especially in the G2C model, authentication and privacy are the two major worries of the government and the citizen respectively. We have to put some security protocols in order to strengthen the G2C model. Security is still the key factor, no matter which model it belongs to.

1.3 What This Thesis Contains

This thesis emphasizes on the topic of electronic commerce, especially in a technical

point of view. We are going to discuss the cryptographic aspect of e-commerce protocols.

This thesis is organized as follow. In Chapter 2, cryptographic primitives such as symmetric cipher, asymmetric cipher, digital signature, message digest, zero-knowledge proof, digital certificate, RSA Public Key Cryptosystem, Elliptic Curve Cryptosystem, Blind Signature and the Cut-and-Choose Protocol are explained. In additional to the theory, the trend of e-commerce is also described in Chapter 3. We also discuss the different kinds of payment system used in e-commerce.

In Chapter 4, we discuss about smart card. The nature and the advantage of smart cards are discussed. We also introduce Java Card. Through this chapter, we can obtain a more clear picture of smart card, which is used in one of our proposed system.

In Chapter 5, the first proposed application is discussed. It is a smart card system using elliptic curve cryptosystem as the public key algorithm. Various security features are included, such as multiple certificates system.

Besides smart card, e-cash is another research area of my studies. We give a brief introduction of e-cash in Chapter 6. In this chapter, the idea of e-cash and the main advantages of e-cash are given. We also briefly describe the basic model of e-cash.

In Chapter 7, we propose a new concept of e-cash: recoverable and untraceable e-cash. It allows the user to get back his lost e-cash, yet at the same time, the e-cash remains untraceable.

There are two publications derived from this thesis. Namely *"Multi-Application Smart Card with Elliptic Curve Cryptosystem Certificate"* derived from Chapter 5 and *"Recoverable and Untraceable E-cash"* derived from Chapter 7.

Chapter 2

Introduction to Cryptographic Theories

Cryptography is a kind of science of keeping the message secret [1]. Cryptography deals with all aspects of secure messaging, authentication, digital signatures, electronic money, and other applications. Hundreds or even thousands years ago, cryptographic systems were employed in military organizations to keep message secret. At the end of the 20^{th} century, cryptologies were used in global communications and commercial applications. Nowadays, people like to do their business in the electronic domain. In the world of electronic commerce, people heavily employ dozens of cryptographic techniques and methodologies in order to attain the security, one of the very important elements of electronic commerce.

In this chapter, we will talk about the fundamental theories that are involved in applications mentioned in this thesis.

2.1 Six Cryptographic Primitives

At the basic level of cryptography, there are six cryptographic primitives:

2.1.1 Symmetric Encryption

In a symmetric encryption scheme, the encryption key is the same as the decryption key. A message can be decrypted only if the key matches the encryption key. That is,

$$E_k(P) = C$$

$$D_k(C) = P$$

$$D_k(E_k(P)) = P$$

Where P = plaintext

C = ciphertext

k = encryption / decryption key

In order to do encryption, two parties should get an agreement on the encryption key before they start their communication.

The most famous examples of the symmetric encryption include Data Encryption Standard (DES) [2, 3], Triple DES [4] and the International Data Encryption Algorithm (IDEA) [5].

2.1.2 Asymmetric Encryption

In an asymmetric encryption, the encryption key is different from the decryption key. That is,

$$E_{k1}(P) = C$$

$$D_{k2}(C) = P$$

$$D_{k2}(E_{k1}(P)) = P$$

such that $k_1 \neq k_2$. It is important that the decryption key cannot (practically) be

derived from the encryption key. Each user has a key pair – private key and public key. The private key should be kept secret while the public key should be published and let other people know. If Alice wants to send an encrypted message to Bob, she has to use Bob's public key to encrypt the message. When Bob receives the encrypted message, he has to use his private key to decrypt it. It can also be used to do key exchange or key agreement. After getting the key agreement, users can use symmetric encryption to communicate.

Asymmetric encryption is also known as public key encryption. Examples include RSA [6] and Elliptic Curve Cryptosystem (ECC) [7, 8]. We will discuss the detail algorithm of RSA and ECC in chapter 2.2 and 2.3.

2.1.3 Digital Signature

Digital Signature is similar to handwritten signature, but instead of signing a paper document, digital signature is used to sign an electronic document. A digital signature should contain the following properties: [9]

1. Authentic. Signature convinces the recipient that it is signed by the signer.
2. Unforgeable. Only the signer can sign, no one else can sign the document.
3. Not reusable. The signature is part of the document, and therefore cannot move to another different document.
4. Unalterable. After signing the document, the signer cannot alter.
5. Cannot be repudiated. After signing the document, the signer cannot later claim that he did not sign it.

If Alice wants to send a signed message to Bob, she has to use her private key to sign the document. When Bob received this document together with the signature, he has to use Alice's public key to verify it.

We will talk about different kinds digital signature schemes in chapter 2.5.

2.1.4 Message Digest

It is also know as one-way hash function. The purpose of a hash function is to produce a "fingerprint" of a file, message, or other block of data. In order to fulfill this purpose, a hash function H(x) must have the following properties:

1. H(x) can be applied to a block of data of any size.
2. H(x) produces a fixed length output no matter the size of the input data.
3. H(x) is easy to compute for any given x.
4. It is a one-way function. That is, it is computationally infeasible to find x given h such that $H(x) = h$.
5. It is a collision-resistant function. That is, it is computationally infeasible to find $y \neq x$ such that $H(x) = H(y)$.

The first three properties ensure the hash function can be put into practical application. The forth property ensures no one can guess the message from the message digest. The last one guarantees that an alternative message hashing to the same value as a given message cannot be found. If many different messages map to one single value, this hash function loses its significance.

It is mainly used in producing digital signature. The two famous hash functions are Secure Hash Algorithm-1 (SHA-1) and MD-5. The former hashing algorithm produces a 160-bit hash-value while the latter algorithm produces a 128-bit output.

2.1.5 Digital Certificate and Certificate Authority

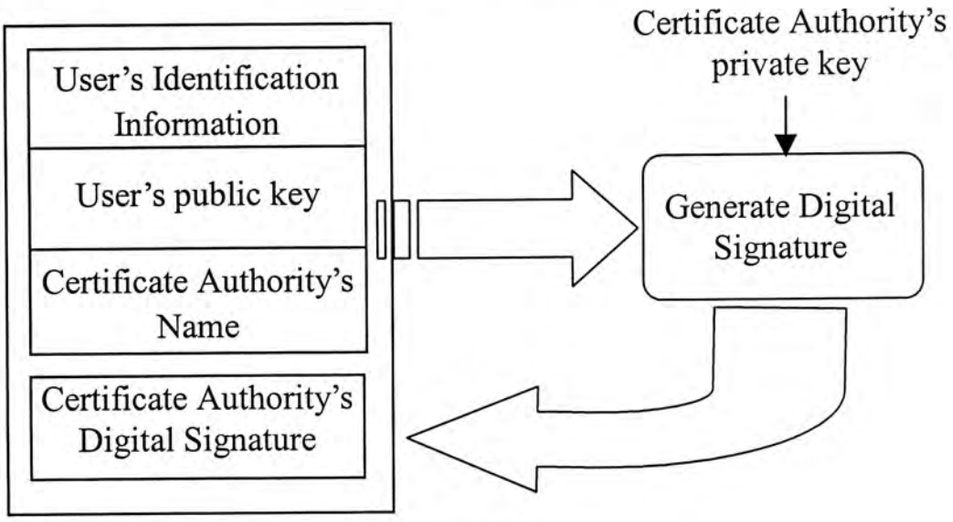

Figure 2.1

The above figure (figure 2.1) shows the structure of a digital certificate. The certificate contains user's personal information, his public key and most importantly, the Certificate Authority's signature. The Certificate Authority (CA) is a third party that can be trusted, such as Microsoft, VISA and MasterCard. If anyone wants to receive data from his friend, he only needs to present his certificate to his friend, then his friend can get the public key which is then used to encrypt the data.

In a large network system such as Internet, it is difficult to find your friend's public key. Without his public key, you cannot use the public key algorithm to communicate securely with him. If you ask him to send the public key to you, it is possible for hackers to "steal" his public key, instead sending the hacker's own public key to you. Therefore it is also not secure.

Now instead of just sending the public key, your friend sends his certificate to you. You can then extract his public key from the certificate. This is a secure communication because although someone may also "steal" the certificate, he cannot make his own certificate because the CA's signature is required for a certificate. (The CA's public key can be found anywhere, or is embedded in famous software such as Windows.)

2.1.6 Zero-Knowledge Proof

The certificate alone cannot confirm that the individual presenting the certificate as proof of identity is actually the right owner. Someone can pretend he is your friend and using your friend's information to ask the CA to give him a certificate. Therefore zero-knowledge proof is needed for authentication. It is also known as random challenge. It is used to authenticate the ownership of a user. The main idea is to prove that the user owns his private key without revealing it. The algorithm is as follow:

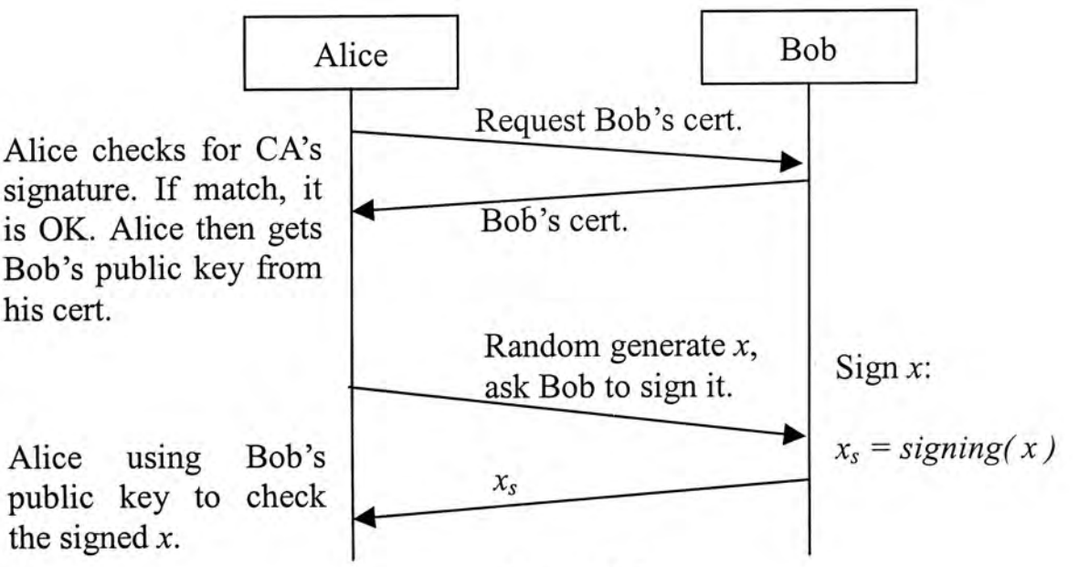

Figure 2.2

In this way, Bob can ensure Alice contains her own private key, that is, to authenticate the ownership of Alice's certificate, yet she does not reveal her private key to Bob.

2.2 The RSA Public Key Cryptosystem

The RSA Public Key [6] Cryptosystem is a widely used algorithm in modern cryptography. It is named for its inventors, Rivest, Shamir and Adleman. The algorithm is as follow:

1. First choose two large prime numbers p and q of equal length (about the same size).
2. Calculate $n = pq$.
3. Randomly choose an integer e such that $gcd(e, (p-1)(q-1)) = 1$.
4. Use extended Euclidean algorithm to find out an integer d such that

$$de = 1 \ (mod \ (p-1)(q-1)) \qquad \text{or} \qquad d^{-1} = e \ (mod \ (p-1)(q-1))$$

5. The number n and e are the public-key, and the number d is the private key.
6. To encrypt a message m , first divide it into a number of blocks smaller than n .
7. The encryption formula is just:

$$c = m^e \ (mod \ n)$$

8. To decrypt a message, only need to calculate:

$$m = c^d \ (mod \ n)$$

9. This algorithm works because:

$$c^d = (m^e)^d \ (mod \ n)$$

$$= m^{e\,d} \ (mod \ n)$$

$$= m^{1+K(p-1)(q-1)} \ (mod \ n)$$

$$= m\,(1)^{K(q-1)} \ (mod \ p) \quad = m\,(1)^{K(p-1)} \ (mod \ q)$$

(by Fermat's Little Theorem: $a^{p-1} = 1 \ (mod \ p)$)

$$= m \ (mod \ pq) = m \ (mod \ n)$$

(by Chinese Remainder Theorem:

If $x = y \ (mod \ p)$ and $x = y \ (mod \ q)$

then $x = y \ (mod \ pq)$)

Therefore, the original message is recovered by this formula.

The security of RSA is based on factorizing the large number n.

2.3 The ElGamal Public Key Encryption Scheme

The security of the ElGamal Public Key Encryption Scheme [24] is based on the difficulty of the discrete logarithm problem. The basic ElGamal encryption scheme is described as follow:

1. First choose a prime number p and a generator α in Fp.
2. Select a random integer a, such that $1 \leq a \leq p\text{-}2$, and compute $\alpha^a \pmod p$.
3. The public key is (p, α, α^a) and the private key is a.

If Bob wants to encrypts a message m to Alice:

1. Obtain Alice's public key (p, α, α^a).
2. Choose a random integer k, such that $1 \leq k \leq p\text{-}2$.
3. Compute $\gamma = \alpha^k \pmod p$ and $\delta = m\,(\alpha^a)^k \pmod p$.
4. Send the ciphertext $c = (\gamma, \delta)$ to Alice.

When Alice receives the message and wants to decrypt it:

1. Use her private key a to recover m by computing $(\gamma^{-a})\,\delta \pmod p$.
2. It works because

$$(\gamma^{-a})\,\delta = (\alpha^{-ak})m(\alpha^{ak}) = m \pmod p.$$

2.4 Elliptic Curve Cryptosystem (ECC)

Elliptic Curves were first suggested in 1985 by Victor Miller [8] and Neal Koblitz [7] for implementing public key cryptosystem. The points on an elliptic curve E over a finite field K form an abelian group. The addition operation of this group is easy to implement. Moreover, the discrete logarithm problem in this group is believed to be very difficult, even much harder than the discrete logarithm problem in finite fields of the same size as K [10].

Elliptic curves take the general form of the equation:

$$y^2 + axy + by = x^3 + cx^2 + dx + e$$

where a, b, c, d and e are real numbers satisfy some conditions which depends on the field it belongs to, such as real number or finite field. There is a point O called the point at infinity or the zero point. The basic operation of elliptic curve is addition. The addition of two distinct points on elliptic curve can be illustrated by the following figure [6]:

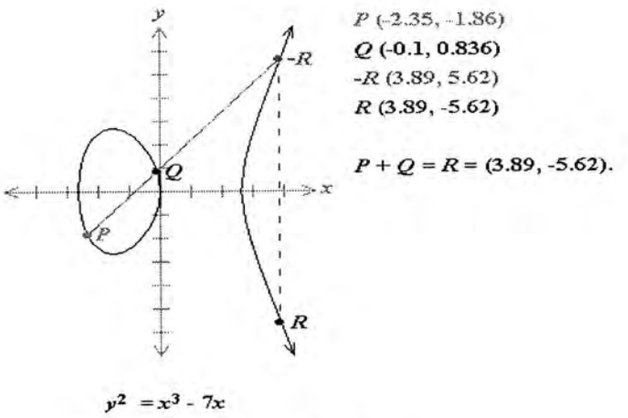

Figure 2.3

To double for a point P, it is equivalent to do $P + P$. Similarly, we can calculate $3P = 2P + P$ and so on. One important property is that it is very difficult to find an integer n such that $nP = Q$.

2.4.1 The Algorithm of Elliptic Curve Cryptosystem

In order to use elliptic curve to do cryptographic operation, some basic setup is needed:

1. Find a curve $y^2 = x^3 + ax + b$ over Fq. (or over $F2^m$)
2. Find a point $G = (x_o, y_o)$ such that Order(G) = p which is a large prime number. (Order(G) is defined as p such that $pG = O$)
3. The Curve(C) and point G is known to everyone and can be shared by multiple user.
4. Private key is an integer: s such that $1 < s < p$.

5. Public key is a point on the Curve: S where $S = sG$.

The security of ECC is based on how difficult to determine k given kP and P. This is referred to as the *elliptic curve discrete logarithm problem* (ECDLP). Based on the ECDLP, various kind of operation can be executed, such as encryption/decryption, key ex-change and digital signature.

2.5 Different kinds of Digital Signature

There exist many different kinds of digital signature. Here we are going to introduce those derived from RSA and ECC algorithms.

2.5.1 RSA Digital Signature

The RSA digital signature scheme is similar to the RSA encryption/decryption algorithm. Let d be the private key and e be the public key. The signature s on a message M using the private key d is

$$s = M^d \ (mod\ n)$$

To verify the signature, just need to check for the following equality using the public key e:

$$s^e = M \ (mod\ n)$$

2.5.2 Elliptic Curve Nyberg-Rueppel Digital Signature

We are going to introduce an ECC digital signature scheme. In fact, it is just the Elliptic Curve version of the original Nyberg-Rueppel Digital Signature. The procedures are as follow:

1. Setup all the parameters needed for ECC: the curve C, common point G, private

key s, public key S and modulus q.

2. Let the message needed to be sign be f.

3. Randomly generate a key pair (u, $V = uG$). Let $V = (x_v, y_v)$.

4. Calculate $c = x_v + f \ (mod\ q)$

5. Calculate $d = u - sc \ (mod\ q)$

6. Output the pair (c, d) as the signature.

The Signature Verification process is as follow:

1. Calculate $P = dG + cS$ (c, d are received from the sender, G is the common point on the curve, S is the public-key of the sender.) Let $P = (x_p, y_p)$.

2. Calculate $f' = c - x_p \ (mod\ q)$

3. Accept signer's signature for message if and only if $f' = f$.

2.6 Blind Signature

Blind signature is a kind of digital signature such that the signer knows nothing about the message to be signed. The first blind signature is invented by David Chaum [13]. It is based on the RSA algorithm.

Suppose Alice wants to obtain a blind signature from Bob for the message m. Let the private key of Bob be d and public key be e using modulus n. The algorithm is as follow:

1. Alice choose a random integer r.
2. Calculate

$$M' = r^e M \ (mod\ n)$$

3. Bob signs on M' using his private key d:

$$M'^d = (r^e M)^d \ (mod\ n)$$

and returns to Alice.

4. Alice unblinds M'^{d} by dividing M'^{d} with the divider r.
5. An ordinary signature on M is obtained:

$$M'^{d} / r = (r^{e}M)^{d} / r = M^{d} \ (mod\ n)$$

Using this algorithm, Alice can obtain an ordinary signature on the message M and at the same time, Bob cannot read M because it is blinded by the blinding factor r. But in this way, Alice can cheat easily by giving a false message to Bob for the signature. Therefore we have to use another technique called *cut-and-choose* protocol [14] to protect Bob from cheating. We will discuss this protocol in the following section.

2.7 Cut-and-choose protocol

It is known as cut-and-choose mainly because it is similar to a classic protocol of dividing a cake fairly:

1. Alice cuts the cake in half.
2. Bob chooses one of the halves for himself.
3. Alice takes the remaining half.

This is the best way to obtain a fair result because Bob can choose whichever half he wants. By using the cut-and-choose protocol, the security problem stated above for the blind signature is solved. Bob can be protected from being cheated by Alice [15]. The protocol is explained as follow:

1. Alice chooses k random integers r_i for $i = 1$ *to* k.
2. She has to calculate:

$$M'_i = r_i{}^{e}M \ (mod\ n)$$

and submit all the k M'_i to Bob.

3. Bob randomly choose an integer j such that $1 \leq j \leq k$ and asks Alice to show

the value of r_i for all $i = 1$ *to* k except j.

4. Bob then verifies whether $M'_i = r_i{}^e M \ (mod\ n)$ for all $i = 1$ *to* k except j.
5. If the verification is passed, Bob signs the unopened message M'_j and returns it to Alice.
6. Alice unblinds M'_j by dividing M'_j using r_j as the divider.
7. Alice finally obtain a signature on M because

$$M'_j{}^d / r_j = (r_j{}^e M)^d / r_j = M^d \ (mod\ n)$$

In this way, we can see that Alice can cheat successfully only with the probability of $1/2^k$. It can be reduced by increasing the value of k.

2.8 Diffie-Hellman Key Exchange

This is the first public-key algorithm invented [16]. Difference from RSA, it cannot be used to encrypt and decrypt messages. Instead, Alice and Bob can use this algorithm to exchange their secrete key and then they can use symmetric encryption for their secrete transmission. The algorithm is as follow:

1. First Alice and Bob have to agree on two integers, n and g, such that $1 < g < n$.
2. Alice chooses a random integer x and calculate:

$$X = g^x \ (mod\ n)$$

3. Bob also chooses a random integer y and calculate:

$$Y = g^y \ (mod\ n)$$

4. Alice keeps x secrete and sends X to Bob. Bob keeps y secrete and sends Y to Alice.
5. Alice computes $k = Y^x \ (mod\ n)$ and Bob computes $k' = X^y \ (mod\ n)$.
6. We can note that $k' = k = g^{xy} \ (mod\ n)$. This is the common secrete derived by Alice and Bob. They can use this secrete to do symmetric encryption.

Chapter 3

Introduction to E-commerce, M-commerce and Rich Media M-commerce

Thousands years ago, people performed their trading activities by exchanging goods and services for other goods and services. However, this transaction method requires a double coincidence of wants. Therefore, after the invention of money, people began to use money as the media of doing business.

The earliest money was the commodity money which included items such as seashells, tea and even tobacco. Later expensive metals such as gold and silver became the materials of money. In the modern world, paper money became more general.

In the digital world nowadays, people like to move their business from the paper domain into the electronic domain. Various kinds of electronic payments systems are introduced, such as electronic money, credit card and debit card. These kinds of payments enhance the use of Internet or even the wireless mobile network to conduct

electronic commerce.

In this chapter, we give a brief description of the trend of electronic commerce and some of the common payment systems used in doing e-commerce.

3.1 1st Generation of E-commerce

In the 1st generation of e-commerce, people use their personal computers to go to the World Wide Web (WWW) to conduct commercial activities. They can go shopping from websites. They can also buy services from the web. They can even process financial transaction using the WWW, such as e-banking service from many banks.

If people buy things or service from the web, they usually have to give their credit card number to the merchant through the web. If people want to make some transaction for their bank account, they even have to give their bank account password to the bank through the web. Try to think that if these information are captured by other people. They can pretend the owner to go shopping or even worse, transfer all the money from the owner's bank account to the hacker's back account. It seems very dangerous to do e-commerce without any security protection.

Therefore, we can see the security is one of the main concerns of e-commerce. Solutions include adding some security protocol to the existing network, such as Secure Socket Layer (SSL) [18] and Secure Electronic Transaction (SET) [19]. Public Key Infrastructure (PKI) including digital certificate should be used in order to support these protocol.

3.2 2nd Generation of E-commerce – M-commerce

During these years, the popularity of mobile phone has been increased rapidly. About 10 years ago, people only used their mobile phone for voice communication. But

recently, as more mobile phones support data communication, more people try to use data communication services conducted by their mobile phones. This produces a suitable environment of doing commercial activities using mobile phones. We call this kind of commercial activities "mobile commerce" (or just for simple, "m-commerce").

The first kind of m-commerce activity is mainly based on SIM Tool Kit which can be regarded as the enhanced version of Short Message Service (SMS). The main service provided is mobile banking service. Customer can do their bank account transaction or enquiry through their mobile phone. But there are some restrictions, for example, each content provider (such as bank) has to make an agreement with the mobile service provider in order to let the subscribers to use SIM Took Kit to access the services provided by each content provider. When comparing to the WWW, it seems not to be a good choice of doing commercial activities.

Since the birth of the first WAP phone, Nokia™ 7110 [19], a new media of doing m-commerce has been created. Wireless Application Protocol [20] (WAP) is similar to WWW. The main difference is that WAP is decided specific for mobile phones while WWW is for computers. Users can go to any URL they want to browse. It is an interactive communication. Therefore, many m-commerce applications can be developed based on WAP.

Yet m-commerce developers face the same problem as e-commerce – security problem. In fact, it is much more difficult to deal with it for WAP because the processing power of a SIM card or a mobile phone is much less than a personal computer. It is very hard to do a 1024-bit encryption by just using the SIM card. Furthermore, implementing PKI is also not an easy job in the mobile world. In fact, WAP version 1.1 does not support certificate system. The only encryption layer is Wireless Transport Layer Security (WTSL). Certificate system is only supported from

WAP version 1.2 beyond.

3.3 3rd Generation of E-commerce – Rich Media M-commerce

As the 3G mobile service will come to market this year, there is a revolution on the wireless data communication. High bandwidth provides a multimedia way of communication. Not only text, but also graphics, sound or even video can be transmitted. For the handheld device, not only mobile phones will be used. PDAs will become a mobile device also.

As both the bandwidth and the computing power of the mobile device will be highly increased, we can foresee that more applications that required high bandwidth and high security level can be implemented using the 3G technologies. Commercial activities can be enhanced into Rich Media Mobile Commerce based on the 3G foundation.

3.4 Payment Systems used in E-commerce

Nowadays there are various kinds of payment systems used in e-commerce. We are going to introduce some of the most common payment systems.

3.4.1 Electronic Cash

This is the digital analogy to paper cash. It is an offline payment system, that is, no connection to the bank is required when doing electronic cash transaction. The main advantage of it is the untraceability of the e-cash. Unlike other payment systems, the receiver of the e-cash does not know whom the e-cash comes from. Neither the bank nor the merchant knows about the owner of the e-cash. Yet when a user double spends

the e-cash, the bank can catch him.

We will talk more about e-cash on chapter 6.

3.4.2 Credit Card

This is the most common payment method used in doing e-commerce. The two major credit card companies include VISA International and Mastercard which makes up a large number of member banks.

Credit Card is an online payment systems, that is, a three points connections is required – user, merchant and bank. In a POS counter, the owner of the credit card has to give his signature together with the card in order to prove that he is the owner of the card. But when it is used in the web payment, no signature is required. The owner only needs to type the credit card number as well as the expired date in the payment form.

It is quite insecure because if the credit card number is captured by another person, this person can use that credit card to buy things on the web. Therefore, some security protocol is needed for the protection of the credit card holder, such as SSL [17] and SET [18].

3.4.3 Combined Payment System

In the recently years, smart card technology grows rapidly. Some credit cards are now implemented on smart cards. They can integrate e-cash into it also. VisaCash™ is an example. Some "cash" is rewarded and stored inside the smart card when the user uses this credit card for payment. The user can use this "cash" next time he buys things using this credit card.

Chapter 4

Introduction to Smart Card

"The smart card is one of the latest additions to the world of information technology. Similar in size to today's plastic payment card, the smart card has a microprocessor or memory chip embedded in it . [21]" In this chapter, we are going to discuss about this kind of pocket size computer.

4.1 What is Smart Card?

Smart card is a kind of plastic card which are classified into memory card and microprocessor card. Memory card simply just stores data into it. You can regard it as a small floppy disk. Microprocessor card contains a small processor, memory and an operating system such that it can not only store information, but also execute some programs. Some of them even have some built-in security features. All we are going to discuss here is about the latter – microprocessor smart card.

Smart cards have two different kinds of interface: contact and contactless. Contact Smart Cards need to be inserted into a smart card reader, making physical

contact between the chips of the card and the reader in order to do any communications. Contactless Smart Cards contain an embedded antenna which allows the smart cards and the readers communicate in a wireless environment.

4.2 Advantages of Smart Cards

Smart cards become a new device widely used in the information technology world. We are now going to discuss what are the main advantages of using smart cards.

4.2.1 Portable Device

Smart Card is a really portable device. If you think that a notebook computer is already a portable device, then it seems that the smart card can be regarded as a part of your body! You can put it in your pocket and bring it everyday and everywhere. The size and weight are both negligible.

4.2.2 Multi-applications

Unlike a magnetic card, a smart card can store multi-applications instead of just one single application. In the past, you may have to bring many cards such as student identity card, library card, photocopy card, medical card etc. But if smart cards are used to replace these cards, just one single smart card is enough to implement all these kinds of applications. It is much more convenience for user and much more cheaper for the production cost.

4.2.3 Computation Power

As we have stated before, microprocessor smart cards contain a small CPU. It implies that smart cards are able to perform some calculations and programming. We can then

program the smart cards. We can give some instructions to the smart cards. Using this valuable feature, many different kinds of applications can be developed based on smart cards.

4.2.4 Security Features

For each smart card, there is a PIN which is only known to its owner. If the owner loses his smart card, no one else can use his card because they do not know the PIN. Besides, cryptographic algorithm such as encryption can be used in order to achieve a higher security level of the smart card. As stated above, smart cards are capable of doing program such as encryption and decryption so that the stored information can be transmitted without compromising confidentiality.

4.3 What can Smart Cards Do?

Smart cards play an important role in our daily life. Let us observe our surrounding environment and we can easily find out that there are really many occasions that we are using smart cards [22].

1. Wireless Communication – Every GSM mobile phone contains a SIM card. A SIM card is just a smart card without the plastic part!
2. Transportation – Our Octopus card is a good example. It can be used to pay for many different kinds of transportation, such as bus, ferry, MTR, KCR, LRT. Besides, it can be also used to buy drinks in the drinks-automatic-selling machines.
3. Banking and Payment – Mondex Card [23] is an electronic cash smart card. It can store e-cash. We can also transfer some money from our bank account to the Mondex Card.

4. Government ID Card – The new generation of citizen ID card will be a smart card combining identity card, driving license, passport etc. which makes life much simpler.

These are only some examples. You can also find out more and more! This can prove that smart cards have already become part of our personal belongings. More importantly, smart cards are the necessary elements in the e-commerce world.

4.4 Java Card

Java is not only a programming language, but it is also a platform. A Java Card is a smart card which has embedded Java as its own platform. Developers can write Java program into it. It is a new concept of smart card. Now we are going to discuss the main advantages of this new-concept smart card.

1. Software and hardware independent: Java is a software-only platform that runs on top of other hardware-based platforms. In order words, Java program can be run on different kinds of operation system such as Windows98/2000, Linux, MacOS, and Solaris. That means we can communicate a Java Card with a machine running of these operation systems. There is no limitation of operation system on the terminal side.
2. Web Browsers compatible: Web browsers (including Internet Explorer™ and Netscape™) are all java compatible. They are Java Virtual Machines and therefore can communicate with the Java Card. Furthermore, through this communication channel, the smart card can be used to develop application for the Internet.
3. High Level programming language with Object-Oriented Property: Java is a

high level programming language. It is easy to develop and debug. Besides, its object-oriented property allows developers to add further application into the system without re-construct the whole system. This is very important in a fast growing IT environment.

As the Java Card provide a channel for communications between the smart card and the Internet, many network applications can be developed using Java Card. For examples, E-mail security, SET-based Internet shopping and other kinds of cash replacement such as E-cash. Electronic tickets can also be brought from the web and store inside the smart card.

In the next chapter, we will present a smart card certificate system. More specifically, our system employs elliptic curve cryptosystem as our public key algorithm and the smart card we choose is Java Card.

Chapter 5

A New Smart Card Certificate System

One of the main barriers to the development of multi-applications smart card is the limitation of the memory size and processing power. In this chapter, we are going to present a solution to it by using Elliptic Curve Cryptosytem (ECC) as the public key scheme. Based on the ECC, we also employ digital certificates for the whole system. Furthermore, our proposed system also supports multiple digital certificates for a smart card using the "Hyper-Link" concept. This is the foundation of other secure applications, such as e-banking.

Here we first give an introduction and a brief overview of our system in section 1. In section 2, we give a comparison between RSA and ECC, followed by the system architecture in section 3. In section 4 we will analyze on the security of our system. Certain extensions will be discussed in section 5 and there will be the conclusion in section 6.

5.1 Introduction

Recently, smart card plays an important role in our daily life. It is a portable storage media. It is also a pocketsize computer. Multiple applications can be easily implemented into a smart card, even with security features. Anyone who has a smart card will be able to electronically and securely interact with several servers or service providers. We have also talked about the details of smart card in the last chapter.

Security is one of the main concerns in the smart card system and cryptography is the foundation of data security. In the science of cryptography, there are 6 primitives, namely symmetric cipher, asymmetric cipher, digital signature, message digest, zero-knowledge proof and digital certificate. A digital certificate is the highest level among the six. It contains the public key of the owner, which is signed by the Certificate Authority (CA) digitally. In order to verify a certificate, zero-knowledge proof will be used.

However, if your certificate is stored on the computer of your office, then you cannot use it when you are at home. It is very inconvenience. It is more desirable if the certificate is stored in a portable media with processing power. Smart card is the media that can fulfill the above requirement. But the chip on the smart card is restricted in both memory and processing power. It is better to use a public-key cryptography such that it can provide the same level of security with the shortest key length possible. The Elliptic Curve Cryptosystem (ECC) [7, 8] is the most desirable cryptography algorithm to do this task [10]. It can provide equivalent security but with shorter key lengths when compared with other existing public key schemes, such as the RSA Cryptosystem [6] which depends on the difficulty of factoring large integers, or the ElGamal Cryptosytem [24] which depends on the difficulty of computing discrete logarithms over finite fields.

This is the main reason that we employ ECC as the public-key algorithm used in the smart card system. In our system, the private key and public key pair are all generated inside the smart card. It can ensure that the private key never leaves the card. The public key is then transmitted through the card reader to the CA. The CA puts the public key, together with the other information of the user, into a digital certificate. It also signs it digitally. The signed certificate is transmitted through the card reader into the smart card. We will talk about our system in detail in section 3.

Furthermore, our smart card system employs Java as the smart card operation system. Java is a cross platform object oriented language which can increase the portability of our smart card system.

5.2 Comparison between RSA and ECC

The major advantage of ECC over RSA is ECC needs less computation than RSA but it still can achieve the same or even higher level of security. Table 5.1 [12] gives cost-equivalent key sizes. It gives the size, in bits, for equivalent keys. The time to break is computed assuming a machine can break a 56-bit DES key in 100 seconds, and then scaling accordingly.

ECC Key	RSA Key	Time to Break	Machines	Memory
112	430	< 5 minutes	105	Trivial
160	760	600 months	4300	4 Gb
192	1020	3 million years	114	170 Gb
256	1620	10^{16} years	.16	120 Gb

Table 5.1

It can be seen that by using ECC, less bits can be used to maintain the same level of security. It implies that much less computation power is needed for ECC than RSA in order to achieve the same security level.

Implementation of public-key cryptography in a smart card has numerous limitations. The two major limitations are constrained memory and limited computing power. Any addition to memory or processing capacity increases the cost of each card. As we can see from above, ECC needs less computation power, thus it is more suitable than RSA in implementing public-key cryptography in a smart card.

5.3 System Architecture

In this section we are going to describe the system architecture in terms of cryptographic and practical implementation.

5.3.1 System Setup

In our basic system, there is a central Certificate Authority (CA) and a smart card user, Alice, who can be a student of a school, or a staff of a company, or a citizen of a country. In addition to it, there can be also more than one CA. For example, let the government be the central CA and the school can also be another CA. While at the same time, Alice can be the student of the school and the citizen of the country.

Each CA should have its own setup parameters which are as follow:

1. Find a curve $y^2 = x^3 + ax + b$ over Fq. (or over $F2^m$)
2. Find a point $G = (x_o, y_o)$ such that Order(G) = p which is a large prime number. (Order(G) is defined as p such that p$G = O$)
3. CA has its private key s and public key S.
4. The Curve(C), the integer q and the point G are known to everyone and can be shared by multiple users.

After that, the CA is said to be ready and welcome for user to apply for a certificate.

5.3.2 Apply for a certificate

If a user wants to apply for a certificate, the following steps are taken:

1. If Alice wants to apply for a certificate, she has to come to the CA in person. This is needed because the CA needs to verify her identify.
2. Alice has to type her personal information and password into the computer of the CA directly.
3. The CA stores the data into the computer temporary.
4. At this stage, the smart card is going to generate the key pair internally. The process is as follow:
    a. Randomly generate an integer s.
    b. Multiple s by the common point G. The resulting point $S = sG$ is the public key.

    By this way, the private key is never left the smart card. It is the most secure way for generating the private key. (We will discuss the security issues of this in section 4.)
5. After generating the key pair, the card outputs the public key (a EC point) to the CA server.
6. Together with the Alice's information, such as her name, and her public key which is generated by the smart card, the CA uses its own private key to sign it. The signature scheme we use is called Elliptic Curve Nyberg-Rueppel Digital Signature. It works as follow: (which is also described in section 2.5.2).
    a. Let $f=(PublicKey_{Alice}, Information_{Alice})$ which is going to be signed.
    b. Randomly generate a key pair (u, $V = uG$). Let $V = (x_v, y_v)$.
    c. Calculate $c = x_v + f \ (mod \ q \ )$ and $d = u - sc \ (mod \ q \ )$. ($s$ is the private key of CA.)

d. Output *(c, d)* as the signature. That is,

$$Sign_{CA}(PublicKey_{Alice}, Information_{Alice}) = (c, d)$$

7. The certificate $\{PublicKey_{Alice}, Information_{Alice}, (c, d)\}$ is then formed. After generating the certificate, the CA copies the certificate into the smart card.

5.3.3 Authentication of Alice

Most of the applications need to authenticate a user. For example, if Alice wants to go into her lab which only allow students of her department to go inside. Here is the procedure for the authentication of Alice:

1. Alice presents the smart card to the terminal.
2. She has to type her PIN in order to get the access.
3. If the PIN is correct, the terminal asks for her certificate.
4. Alice (her smart card) sends the certificate to the terminal and it verifies it using CA's public key. The process is as follow:
    a. Calculate $P = dG + cS$ (c, d are received from Alice, G is the common point on the curve, S is the public-key the CA.) Let $P = (x_p, y_p)$.
    b. Calculate $f' = c - x_p \ (mod\ q)$
    c. Accept it if and only if $f' = f$.
5. The terminal randomly generates an integer x and requests Alice (her smart card) to sign it using her own private key.
6. Alice (her smart card) signs it and sends it back to the terminal.
7. The terminal verifies it using Alice's public key extracted from her certificate. (the signature algorithm is described in section 2.5.2).
8. The terminal will only accept Alice if she can pass the verification.

The verification process can be illustrated by the following diagram:

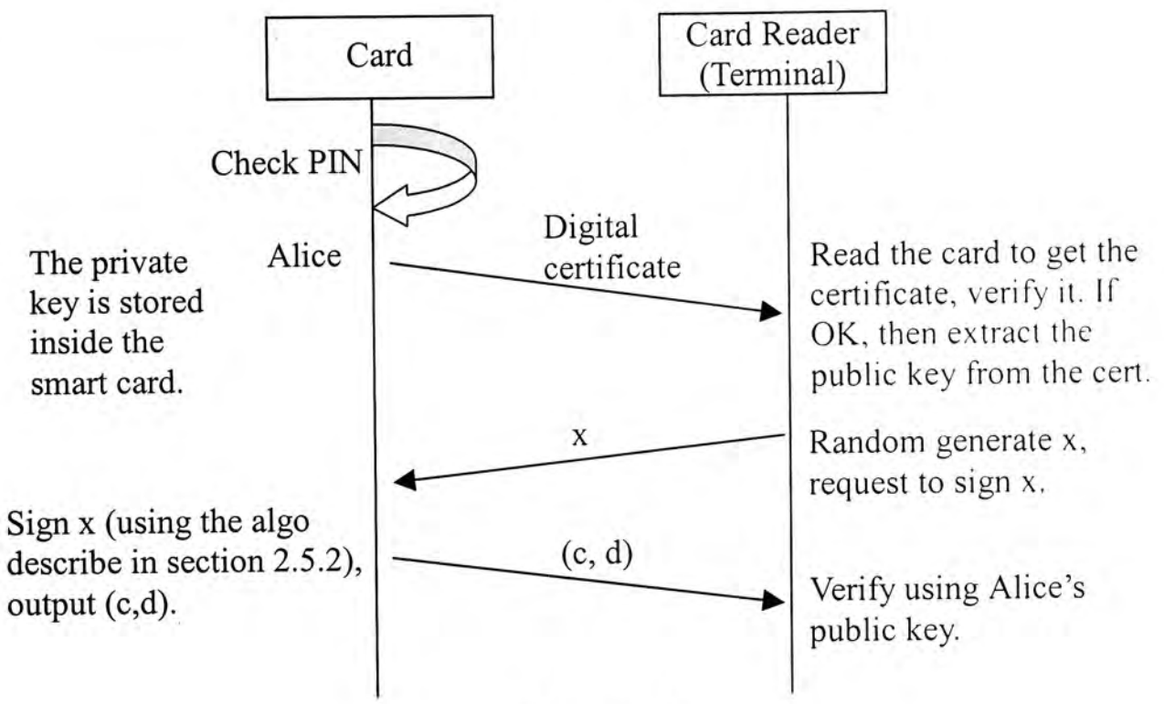

Figure 5.1

This is called the zero-knowledge proof because Alice can prove she knows her private-key without revealing it.

5.3.4 Other Certificates – the "Hyper-Link" concept

The main certificate (for example, given by the government) of Alice is stored inside the card. Then how about her other certificates? Digital certificate is very common and a user may have many certificates. A bank can give certificate to its customer. A school can give certificate to the students. Even a "dot-com" can also give certificate to its user. It is difficult to store every certificate inside the smart card because the memory of a smart card is very limited. In our proposed system, we employ the concept of "Hyper-Link". These other certificates will be stored elsewhere instead of the smart card. These certificates are only needed when you are doing web browsing or on-line shopping. Therefore we put these certificates in a web server. Only the URLs will be stored inside the smart card. If a particular certificate is needed, just go

to the URL and download the required certificate. It is similar to a "Hyper-Link" in the HTML language.

It is very clear that the size of a "URL hyper-link" is extremely small when comparing to a certificate. By this way, even a smart card can store up to a thousand of "URL hyper-link" of certificate. It will be much better to put each certificate into more than one web server. If one of the web servers is out of service, we can still go to other web servers to download the certificate.

5.3.4.1 Generation of the "hyper-link"

It is the same as the process 1 to 6 of section 5.3.2. But for this time, the certificate is not stored on the smart card. Instead, the CA uploads the certificate to its web server(s). It then sends the URL hyper-link(s) of the certificate back to the smart card for storage. We can also notice that although the certificate is stored somewhere else, the private key corresponding to that certificate is still stored inside the smart card.

5.3.4.2 Authentication of Alice using the "hyper-link"

It is almost the same as the authentication process using the main certificate, which is described in section 5.3.3. But after Alice has typed her PIN correctly (after execute process 1 – 3 in section 5.3.3), the terminal reads the hyper-link of the required certificate. It downloads the certificate from it. After getting it, then the remaining process is the same as process 4 – 8 in section 5.3.3.

There is a tradeoff of the "hyper-link". As you can see that the terminal has to download the required certificate from the web, the terminal should have a connection to the Internet. Therefore, off-line terminal cannot employ this mode of service. However, there are still many on-line occasions that allow Alice to use her

"hyper-links", such as on-line shopping on the web.

5.3.5 Multiple Applications

Based on these security modules, we can build up the entire smart card system which consists of many applications. Here we present some of the most important applications in our proposed system.

1. Digital Signature

Digital Signature is similar to handwritten signature, but instead of signing a paper document, digital signature is used to sign an electronic document.

A digital signature is necessary in all the activities in the e-commerce world. When Alice wants to sign a document or send an email with her signature, she has to use her private key to do so. The private key is stored inside the smart card. It never leaves the card. But if the whole document is transferred to the smart card to be signed, it is not practical because of the limited memory of the smart card. Instead of the whole document, we make the message digest of the document using a one-way hash function, such as SHA-1. The size of the resulting data is only 160 bits.

2. Encryption

Sometimes Alice wants to encrypt a document before sending to Bob. Before doing so, Alice has to get the public key of Bob. What she has to do is just asking Bob to send her his own certificate. If he wants to use his main certificate which is stored in his smart card, then he has to send it from his smart card directly. If he wants to use other certificate, he just needs to send Alice the corresponding hyper-link and ask Alice to download from this specific URL.

After getting the public key of Bob, Alice can use it to develop a secret key with

Bob using some key-exchange algorithm, such as Diffie-Hellman key-exchange [16] (described in section 2.8). Then she can use this derived secret key to encrypt the document. At the other side, Bob can also use this secret key to decrypt the document.

3. Online Applications

Java is a cross platform language. More importantly, web browsers can also support Java. It implies that our Java Card can be used compatible with many web-based applications, such as on-line shopping or e-banking service. Furthermore, more applications can be added to the card just by downloading applets into it. The new applets can be put on some web sites and let users to download them through the reader into the smart card. This is a great advantage. Users do not need to bring back their smart card to the card center or administrator in order to add more applications into it. They can do it by themselves at their home or office.

4. Off-line Applications

Although all the off-line applications required the main certificate, there are still many different kinds of applications that can be developed, such as a door lock system (or restricted area access system), self photocopy service, library record and facility booking system. Besides, off-line e-cash protocol such as [25, 26, 27] can be integrated into it as well.

5.4 Security Analysis

Security is a very important matter in all smart card system and it is also what we want to emphasis in our system. We now discuss some of the security issues.

5.4.1 No Crypto-processor is needed

Although there are some smart cards which contain higher level of security features, they have two processors. One is the CPU and the other is the crypto-processor. The price is definitely higher than the general-purpose smart card used in our system. By the way, the US law also restricts the export of these kinds of security products to the area outside US. It is a great barrier to develop a secure multi-purpose smart card system.

We use ECC instead of RSA in our general-purposes smart card system. As a result, the memory and computation power requirement is much lower. Multiple applications which require high security level can be integrated into the same smart card.

5.4.2 PIN Protect

The smart card is PIN protected. If Alice has lost her smart card, other people cannot use it because they do not know the PIN of her smart card. If anyone types the wrong PIN in 3 consecutive times, the card will be locked automatically. Only the central CA can un-lock the card. This is to prevent someone who wants to guess the PIN by trial and error method.

5.4.3 Digital Certificate Protect

Inheriting from digital signature, digital certificate also has the unforgeable property. As the digital certificate is signed by the CA, no one else can forge a certificate by pretending the CA to sign it. Besides, the zero-knowledge proof can prevent other people from stealing Alice's certificate. If Carol has stolen Alice's certificate, she cannot pass the zero-knowledge proof because she does not have Alice's private key.

5.4.4 Private Key is never left the smart card

We can see that in either the process of generation of key pair and the process of authentication of the user, the private key is never left the card. In the encryption and decryption process, we do not directly use the private and public key to do encryption or decryption. Instead we use key exchange algorithm to develop a secret key first. The key exchange algorithm can be taken place inside the smart card. After that, the encryption and decryption process which require a lot of computation power can be done by the computer instead of the smart card. In this way, the private key is still not needed to be given to other computer.

5.5 Extensions

There are several extensions that can be made to our system. Here we are going to discuss some of the major extensions.

5.5.1 Biometrics Security

Some biometrics security features such as fingerprint authentication can be added to the smart card system. The fingerprint of the owner of the smart card can be stored inside the smart card. If an authentication station wants to check the fingerprint of a user, a fingerprint reader is needed. In addition to the digital certificate and the PIN, this is the third level of protection.

5.5.2 E-Voting

Voting using your own computer is not just a proposal. At March, 2000, "*With the click of a computer mouse, Arizona Democrats made U.S. history by becoming the first Americans to use the Internet to cast ballots in a legally binding election.*" [28].

In the election of U.S. President at the end of 2000, there was a dispute of the result. The dispute mainly came from some paper tickets. Using e-voting can surely avoid such argument. According to a survey at the end of December 2000, by Stamford-based Gartner Group, Inc., Americans are generally willing to use Internet- or email-related technologies for voting. [29]

Our proposed system can be extended to support e-voting, not only in the user's own computer, but the user can go to other places to vote, provided that there is a smart card reader attached. It is also 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. We can see that a smart card has the ability to fulfill this requirement: The whole system is a PKI. There is a certificate stored inside the smart card. Furthermore, the smart card also has possessing power to give a digital signature.

Due to the portability of smart card, users can vote wherever they want. They can stay at home. They can stay at office. Even they are not in their own country, they still can vote through the Internet. In fact, it makes no different. This is a great advantage over traditional voting. As a result, using smart card as the e-voting tools, it can be sure that the voting rate will be increased. More importantly, dispute such as those in the U.S. Presidential Election in 2000 will not be happened if using e-voting.

5.6 Conclusion

Smart card is not only a portable storage media, but also a small processing unit. Based on this special property, smart card is a suitable device to develop security system. But one of the main barriers which has been described above is the limitation of the memory and processing power. Using Elliptic Curve can solve the problem.

Besides, as Java is a cross platform language and it can be run by web browsers, it acts as a bridge between the Java smart card and the Internet. Furthermore, the "Hyper-Link" concept allows the user the store many certificate in one single smart card. Based on this security authentication tool, many applications can be developed.

Chapter 6

Introduction to Electronic Cash

Money is a great invention that makes much of modern life possible. People cannot work in an environment without a standard way to trade goods and services. This theory applies in the digital world also. The Internet is just like another convenient way to place an order and complete a transaction. And Electronic Cash is just like the digital analogy of cash.

In this chapter, we are going to discuss about this new form of electronic payment.

6.1 Introduction

The first proposal of using numbers to represent coins was made by David Chaum [13] in 1983. According to his paper, an electronic cash system consists of electronic coins which are numbers. However, the concept of currency that exists in format other than paper and coins has been expressed as early as 1888 in Edward Bellamy's book *Looking Backward* [30]. In that book, cash was replaced by pasteboard and are

hole-punched at each transaction. Until now, there are more than hundreds of proposed electronic cash systems. Some of them are even being used in the real market. We will talk about these products at the last section of this chapter.

E-cash systems generally fall into the following categories: debit-based, credit-based, ATM-based, hybrid, and other innovative models. But here, we focus our attention only on debit-based e-cash protocols.

In an online e-cash system, during a transaction, three parties are involved. Customer-bank or merchant-bank on-line connection is required. In an offline e-cash system, the only parties involved are the payer and the payee. Cut-and-choose is a technique commonly used in early e-cash protocols to achieve untraceability. But it is cumbersome and may be a security liability. Later e-cash protocols use single-term technique where there are only a few terms in a coin. It is more efficient and practical.

6.2 The Basic Requirements

As we have stated above, we can regard e-cash as the digital implementation of paper cash. Therefore its requirements are also very similar to paper cash. According to [26], in the ideal e-cash system, it has six main properties:

1. Independence – The security of the electronic cash cannot depend on any physical condition, such as the hardware protection. It can only depend on itself.
2. Security – The electronic cash cannot be copy, reuse, forged or double spent. Anyone who has processed these cheating activities can be caught.
3. Privacy – The privacy of the user should be protected. That is, no one should be able to trace the relationship between the user and his purchases. User anonymity should be provided. It implies the electronic cash is untraceable.

4. Off-line payment – When a user pays electronic cash to a shop, the only parties involved are the payer and the payee. That is, no customer-bank or merchant-bank on-line connection is required.
5. Transferability – The electronic cash can be transferred from one user to another user.
6. Divisibility – The piece of electronic cash can be subdivided into many pieces such that the total amount of all the pieces should be equal to the amount of the original electronic cash.

In fact, there are only very little e-cash protocols that can fulfill all the six requirements. Most of the protocols are selectively fulfill.

6.3 Advantages of Electronic Cash over other kinds of payment systems

Nowadays there exists many kinds of electronic payment systems. They include credit card, debit card (or called pre-paid card) and e-cash. In this section we are going to discuss the advantages of using e-cash over other kinds of electronic payments.

6.3.1 Privacy

In the credit card payments, user identity is revealed when you give your credit card or credit card number to the shop. The shop has to check the validity of it from the bank. The bank also has the record of each transaction. The information includes the date of each transaction, the amount and the items of things or services the user buys each time, the name of the merchant etc. It seems that the bank knows everything about the credit card and its transaction. No privacy can be maintained. In the 21st century, the request of privacy is increasing everywhere. Therefore, it is better for an

electronic payment that can maintain the privacy of the user.

Electronic Cash has the total advantage of privacy over credit card. When a user uses electronic cash for his payment, it is done in an anonymous way. No one knows the identity of the e-cash. Even the bank does not know who owns the e-cash. Yet if the user does some cheating behaviour such as double spend the e-cash, the bank can find it out and catch the cheating user. In other words, the identity of the e-cash owner will not be reveal unless the owner double spend it. It is just like in geometry, when you are given only one point, you cannot find the equation of the line. But when you are given two points on the same line, you are able to find out the equation of the line. In fact, e-cash protocols also employ this concept.

6.3.2 Off-line payment

If you want to make a credit card payment, there should be an online-connection made between the merchant and the bank. It is needed for checking the validity of the card and the account. It is not convenience because you cannot use your credit card for the places where there is no connection to the bank, such as in the street. In this case, e-cash has a total advantage over credit card. There is no need for any connection made for each transaction of e-cash. The shop only needs to deposit the e-cash at the end of each day, or every two, three... days.

6.3.3 Suitable for Small Amount Payment

Each time when you make a credit card payment, the credit card company will charge you a certain amount of fee, although most of the time, this charges are paid by the shop, instead of the customer. This charge is negligible if you want to buy a large amount of goods or services. However, if you just want to make a few dollars or even fifty cents payment, this charge becomes a large portion when compares with the total

amount you have to pay. Therefore, many shops only allow customers to use credit card payment when they buy more than a certain amount of goods, such as HK$50 or HK$100. If customers just want to buy HK$10 of goods, they are not allowed to use credit card.

There is a kind of electronic cash which are designed for payments of small amount. This is called micropayment. For the design of micropayment protocol, high efficiencies are required so that the transaction cost becomes in-significant. This can be achieved by giving up some of the security requirements such as anonymity. Yet it is very popular since it is much faster and much more convenience. The Octopus of the Hong Kong transportation system is a good example, although it is not exactly micropayment system, but the idea and the concept is the same. Credit card payment is not suitable for the case you pay for the bus fee.

6.4 Basic Model of Electronic Cash

As we have stated before, there are many papers of e-cash protocol. In this section, we choose [15] to give a brief introduction of the e-cash protocol because it is a very significant paper in the history of e-cash. Although it is hardly practical, it builds the foundation of off-line e-cash protocol.

The protocol consists of three phases, namely the cash withdrawal phase, the spending phase and the deposit phase. At the cash withdrawal phase, the bank verifies that the user's identity is embedded in a randomly created coin. This is done in the zero-knowledge manner by using the cut-and-choose method (described in section 2.7). The bank then issues a blind signature (described in section 2.6) on the coin. At the spending phase, the user provides a distinct piece of "information" on his identity, such that one piece of such information provides unconditionally blinding on the

user's identity. However, any two pieces of such information can reveal the identity of the user. At the deposit phase, the shop transfers a payment transcript to the bank. If the bank finds out that there is double spending of a coin, the bank can identify the user because the identity of the user is revealed by the two distinct "information" on it.

6.5 Examples of Electronic Cash

In this section we are going to introduce some e-cash examples that are brought into real markets.

6.5.1 eCash

eCash [31], former named as DigiCash, is formed by David Chaum. Now it is sold to eCash Technologies. It provides business-to-business (B2B), person-to-person (P2P) electronic payments solutions to customers. Besides, it also provides solutions for mobile commerce and beyond. More and more financial institutions join the partnership of eCash.

6.5.2 Mondex

Formally speaking, Mondex [23] cannot be regarded as e-cash because the security of Mondex is depend heavily on the hardware security instead of the algorithm. But the concept is very similar to e-cash, and we can also talk about it here.

It is like a prepaid card. The system operates on a smart card which store the "cash" on a microchip. It also allow divisibility and transferability. You can debit $100 and spend $0.1. You can also transfer $30 into your friend's Mondex card using the transfer machine. Therefore you can regard it as an electronic purse. You can add

money into it in any Mondex-supported ATM machine from your bank account.

6.5.3 Octopus Card

Octopus Card is a kind of prepaid card. It is used in Hong Kong transportation systems, including buses, ferry, MTR, KCR, mini-bus etc. Now you can also use it in buying drinks, dialing telephone and other kinds of services. We choose Octopus Card to be discussed because there is one valuable property that this system has. One of the applications of my thesis is motivated from the idea of this property. It is recoverability.

There are two kinds of Octopus Card. One is general Octopus Card and the other is personal Octopus Card. Recoverability can only be done on personal Octopus Card. If Alice has brought a personal Octopus Card, and has already added $100 into the card. When she has lost it, after reporting and providing her information to the Octopus office, she can get back the money in her lost card.

Yet it cannot be regarded as e-cash formally also. But the idea of the recoverability is highly appreciate.

Chapter 7

A New Recoverable and Untraceable Electronic Cash

Recoverability and untraceability are two obviously conflicting properties. Most of the e-cash papers such as [25, 26, 27] do not include recoverability. Although the e-cash system in [32] contains recovery, it is an on-line e-cash system. In this paper, we introduce a new e-cash protocol which possesses these two properties simultaneously. At the same time, it still remains off-line.

We first give a brief introduction in section 1. The basic idea of our proposed protocol is presented in section 2. In section 3, we are going to give the brief description of S. Brand's Single Term E-cash Protocol [27] in which our proposed protocol is motivated. In section 4, we give the detail explanation of our protocol, followed by the analysis of the security in section 5. Certain Extensions will be discussed in section 6. Conclusion will be done on the last section.

7.1 Introduction

In the ideal e-cash system [26], it has six main properties: Independence, Security, Privacy (Untraceability), Off-line payment, Transferability and Divisibility. Most of the previous papers focus on these areas and provide improvement on these topics. But even using an ideal e-cash system, the user will loss all his e-cash if the computer has crashed and all the files are removed accidentally, or his e-cash wallet, for example, a smart card, has been lost. In the latter case, if someone else finds this wallet, he can spend all the remaining e-cash in a normal way. This situation will not be happened if the user uses his credit card to go on-line shopping. If the user losses his credit card, or recognizes his credit card has been stolen, he can report to the bank so that the credit card becomes invalid at the same moment. But for each transaction the user makes should be online, that is, a three-party connection has to be setup. At the same time the user's anonymity will be lost. Both the shop and the bank can get the identity of the user. We have already discussed it in the last chapter.

In this chapter, we are going to propose a new protocol that can take the advantages of both electronic payments. That is, this is an off-line e-cash system that can support untraceable payment, yet it also supports recoverable. That means even if a user losses his e-cash wallet, after reporting to the bank, he can get back his lost-money. In our scheme, untraceability can be maintained.

Our new system can be regarded as an extension to an existing e-cash system, such as [25, 27]. It is compatible with most of the basic e-cash system, provided that the e-cash is not divisible and transferable.

7.2 The Basic Idea

In this section, we will give the basic idea behind our recovery protocol that allows

users to recover from their lost coins.

If a user losses his e-wallet, or his hard disk gets corrupted accidentally, it is very difficult to find out the exact amount he has lost, unless he has found his lost wallet, or recovered his e-cash files. In our protocol, instead of finding it out directly, we use an indirect method. If we know the total amount of e-cash the user withdrawn from the bank, as well as the total amount he has spent, the difference of these two numbers is the answer that we want to know. In other words,

How much e-cash lost

= How much e-cash withdrawn – How much e-cash spent

Let us have a simple example. If Alice has withdrawn $100 e-cash from the bank, and she has spent $70. Then she lost the remaining e-cash. It can be calculated that she has lost $30 ($100 - $70 = $30).

If we give up untraceability, it is very easy to do so. By using a large database, all the transaction activities are recorded, including the identity of each user and how much e-cash they spend. However, the situation becomes more complex if we want to maintain untraceability. In this case, the bank does not know how much e-cash each user spends, provided that the user does not double spend his money.

In our protocol, we append an additional number to every e-coin. This additional number reveals nothing about the identity of the user. Therefore the anonymity of user can be maintained. If the user lost his e-cash, he only needs to present another number to the bank, then the bank can find out how much e-cash he has spent.

How about if the user pretends he has lost his e-cash, and wants to claim back the money, but in fact he does not loss it? In our protocol, we have a mechanism to prevent this kind of cheating.

The details of this protocol will be described in section 7.4.

7.3 S. Brand's Single Term E-cash Protocol

Our proposed system is motivated from S. Brand's Single Term E-cash protocol [27], therefore hereby we are going to give a brief introduction of his protocol first.

7.3.1 The Setup of the System

The setup of the system consists of B, the bank, U, the customer and S, the shop. The bank also setups two databases: one is called account database which is used to store information about account-holders; the other is called deposit database which is used to store information from deposited payment transcripts.

Let p, q, g, g_1, g_2 be system parameters published by the bank where the order of g, g_1 and g_2 is q. Let x be the secret key of the bank and $h = g^x \bmod p$ be the public key of the bank. Let u_1 be the secret key of the customer and $I = g_1^{u_1} \bmod p$ be the identity of the customer. B computes $z = (Ig_2)^x \bmod p$ and transmits it to U.

7.3.2 The Withdrawal Protocol

If U wants to withdraw a coin, he has to identify himself first, for example, digitally sign a request for withdrawal, or type in the correct password of his account. After that, the following steps are performed:

1. B generates a random number w and sends $a = g^w \bmod p$ and $b = (Ig_2)^w \bmod p$ to U.
2. U generates three random numbers s, x_1 and x_2 and computes $A = (Ig_2)^s \bmod p$, $B = g_1^{x_1} g_2^{x_2} \bmod p$ and $z' = z^s \bmod p$. U also generates two more random numbers u and v and uses them to compute $a' = a^u g^v \bmod p$ and $b' = b^{su} A^v \bmod p$. He also compute the challenge $c' = H(A, B, z', a', b') \bmod p$ where H

is an one-way collision free hash function. After that, U sends the blinded challenge $c = c' / u \ mod \ q$ to B.

3. B sends the response $r = cx + w \ mod \ q$ to U and debits his account.
4. U accepts it if and only if $g^r = h^c a \ mod \ p$ and $(Ig_2)^r = z^c b \ mod \ p$. If it holds, then U computes $r' = ru + v \ mod \ q$.
5. As a result of this protocol, U obtains the following quantities:

$$A, B, sign(A, B)$$

where $sign(A, B) = \{z', a', b', r'\}$. The verification checks the following:

$$g^{r'} \stackrel{?}{\equiv} h^{c'} a' \bmod p$$

$$A^{r'} \stackrel{?}{\equiv} (z')^{c'} b' \bmod p$$

$$c' \stackrel{?}{=} \mathrm{H}(A, B, z', a', b')$$

7.3.3 The Payment Protocol

When U wants to spend his coin at shop S, the following protocol is performed:

1. U sends $A, B, sign(A, B)$ to S.
2. If $A \neq 1$, then S computes the challenge $d = H_o(A, B, I_S, date/time)$, where date/time is the number representing date and time of the transaction and I_S is the identity of the shop S. H_o is another one-way collision free hash function. S sends d to U.
3. U computes the responses $r_1 = d(u_1 s) + x_1 \bmod q$ and $r_2 = ds + x_2 \bmod q$ and sends them back to S.
4. S accepts if and only if $sign(A, B)$ is a signature on (A, B) and $g_1^{r_1} g_2^{r_2} \ mod \ p = A^d B \ mod \ p$.

7.3.4 The Deposit Protocol

After some time (for example, at the end of each day), S sends B the payment transcript, consisting of A, B, $sign(A, B)$, (r_1, r_2) and date/time of the transaction. The following protocol is performed by the bank:

1. If $A = 1$,the B does not accept the payment transcript.
2. Otherwise, B compute d by itself using the information provided by the shop.
3. B verify $g_1^{r_1} g_2^{r_2} \bmod p = A^d B \bmod p$ and $sign(A, B)$ is a signature on (A, B). If not both valid, then B does not accept the payment.
4. B then searches its deposit database to find out whether A has been stored before.
5. If A has not been stored before, then B stores $(A, data/time, r_1, r_2)$ in the deposit database deposited by S and credits the account of S.
6. If A has been deposited before, that means a fraud has been occurred. If the previous transcript was deposited by S and the date/time are the same as the newly deposited transcript, that means S is trying to deposit the same transcript twice. Otherwise, the coin has been double-spent.
7. If the coin has been double-spent, B can use the following equations to find out the identity of the user who has double-spent:
    a. Let (d, r_1, r_2) be the newly deposited payment transcript and (d', r_1', r_2') be the previous deposited payment transcript.
    b. Compute $\frac{r_1 - r_1'}{r_2 - r_2'} = \frac{u_1 s(d - d')}{s(d - d')} = u_1$
    c. Compute $I = g_1^{u_1} \bmod p$ which is the identity of the user who has double-spent.

7.4 The Proposed Protocol

In this section, we will describe our protocol part by part, namely the withdrawal protocol, the payment protocol, the deposit protocol and the recovery protocol. We assume Alice is the user that has lost her e-cash and wants to get recovered. Here we use [27] as our basic e-cash system as well as the notation, which we have described in the last section already.

7.4.1 The Withdrawal Protocol

In our protocol, there are totally 4 parties involved, namely the Bank, user Alice, the shop and the Trusted Third Party (TTP). After Alice has withdrawn e-cash from the bank, she got the coin $\{A, B, sign(A, B)\}$, after executed the protocol described in section 7.3.2. Then the following protocol is performed:

1. She goes to the TTP to get an additional number, x_i, for each coin, for $1 \leq i \leq n$. (We assume there are total n coins.) This additional number has the following properties:

   $$H(x_1) = H(x_2) = H(x_3) \ldots\ldots = H(x_n) = y$$

   where H( ) is an one way hashing function, such as MD-5 or SHA-1, and n is the number of coins withdrawn by Alice at this time.
2. The TTP should maintain a list that records down all the serial number of the coins (in this case, it is the number A) which have been processed. Every time when a user presents his coins, the TTP should check whether these coins have been processed or not. If the list contains the serial number of the coin, the TTP should terminate the process.
3. If the coin is not in the list, the TTP then checks the signature $sign(A,B)$ on (A, B). If it is a valid signature, it gives another signature for each coin. The

signature should include the additional number. Let $S_c = Sign_{TTP}\{ A, B, Sign(A, B), x_i \}$. S_c should be attached to the coin. Now the coin contains the following items: $\{ A, B, Sign(A, B), x_i, S_c \}$.

4. The TTP gives another signature on y and n. Let $S_b = Sign_{TTP}\{ y, n \}$. S_b will be given to Alice. Alice should keep $\{S_b , y, n\}$ in a save place. They are used to recovery her lost coins.

Also note that although the TTP records down the serial number of the coin, the coin is still untraceable because the TTP knows nothing about the owner of the coin.

7.4.2 The Payment Protocol

When Alice wants to spend her coin at the Shop, the following payment protocol is performed:

1. The basic payment protocol based on [27], also described in section 7.3.3.
2. The shop checks also the signature of the TTP. That is, to check if S_c is a valid signature of $\{ A, B, Sign(A, B), x_i \}$.
3. The shop hashes the x_i of the coin to produce its hash value. Then it checks whether the coin is in the *blacklist* (A blacklist is a list containing the *hash value* of the x_i of all the lost coins.). The shop accepts the transaction only if it is not in the blacklist.

7.4.3 The Deposit Protocol

At the end of each day, the shop sends the payment transcript to the bank. The transcript should contain x_i also. This is similar to the deposit protocol of [27], which is also described in section 7.3.4. In additional to the checking procedure, the bank further checks whether the hash value of x_i of each coin is in the blacklist. (The bank should also maintain a blacklist.). The bank does not accept any coins with their hash

value in the blacklist.

7.4.4 The Recovery Protocol

In the previous sub-section, we have mentioned the withdrawal, payment and deposit protocol. Here we will present the most important part of our system, the recovery protocol.

If Alice has lost her remaining coins, she has to do the following:

1. She has to reveal her identity to the bank. She also has to present the back-up number $\{ S_b , y, n \}$ to the bank.
2. The bank checks the signature of S_b on $\{y, n\}$.
3. If the signature is valid, the bank looks up its database to find out all the coins with their hashed values of x_i are equal to y. The maximum number of such coins should be n. These coins are those Alice has already spent.
4. The bank can calculate the difference D between the total amount Alice has withdrawn and the total amount she has spent. D is the amount that Alice has lost.

In order to prevent Alice pretend losing the coins but in fact she does not, the following steps must be taken immediate after she has reported her lost coins to the bank:

1. The bank adds the number y to the blacklist and broadcast it to all the shops.
2. The shop adds this y to the blacklist. If any customer uses any coin with the hashed value of x_i is equal to y, the shop should not accept this transaction because this coin has been either lost or stolen.

7.5 Security Analysis

In this section we analyze on the security of our system.

7.5.1 Conditional Untraceability

Untraceability is maintained in our protocol. We can see that neither the TTP nor the bank knows about the identity of the coin. If Alice does not double spend, her identity is never revealed. However, if Alice looses her coins and wants to get recovery, she has to reveal her identity to the bank. But it is the one and only one case that the user's untraceability is lost provided that the user does not double spend.

7.5.2 Cheating

Alice may cheat by herself or with a dishonest shop in the following ways:

Case 1: Alice may claim that she has lost her coins, yet she does not.

Our protocol cannot prevent this happen. But when Alice gets back her recovered coins, she cannot spend her original coins, because they are all blacklisted.

Case 2: The dishonest shop may hold the coins of Alice, and does not deposit them until Alice has reported she has lost her coins. It seems that Alice can get more than she has lost.

But in fact it is not the case. If the shop delays the deposit of the coins, the bank will not accept them because they have already been blacklisted.

7.6 Extension

Based on the above basic system, there can be some extension to improve the system

in different ways:

Extension 1:

The system can be extended to a more practical manner. As the cost of finding those x_i such that

$$H(x_1) = H(x_2) = H(x_3) \ldots\ldots = H(x_n) = y$$

is not low, the TTP may charge Alice. At the same time, Alice can also choose not to "buy" such insurance service. In this case, the TTP just put 0 for all the value of x_i , y and n. No more modification is needed. By doing this way, the system become more economical and reasonable.

Extension 2:

As finding the x_i is quite time consuming, the TTP can produce many groups of $x_{i,j}$ such that

$$H(x_{1,j}) = H(x_{2,j}) = H(x_{3,j}) \ldots\ldots = H(x_{n,j}) = y_j \,.$$

When j^{th} customer come, the TTP can give the j^{th} group number to him. This is just like to do some pre-calculation and it can greatly save the time of a customer spent in the TTP.

Extension 3:

As the output of the hash function is a constant length integer, there is a finite number of outputs. After all the output integers have been used up (or nearly used up), the system (or the bank) needs to renew all the coins.

7.7 Conclusion

We have presented an e-cash protocol that can support both untraceability and recoverability. Certain tradeoffs are necessary, as discussed above. Our protocol has been motivated by the single-term offline untraceable e-cash protocols in [25, 27]. We have made significant innovations in order to compromise the conflict between untraceability and recoverability. There still remains many open topics for future research, such as how to support divisible and transferable e-cash.

Chapter 8

Conclusion

In this thesis, two aspects of e-commerce are discussed: security of e-commerce and payment method used in e-commerce. We started from the brief describing of the development of e-commerce: the 1st generation is PC-based, the 2nd generation is mobile-based, which is called m-commerce, and the 3rd generation is 3G-based, which is called rich media m-commerce. But we can see that in any stage of the development, security is also the main concern. In Chapter 4, we looked at the smart card at a detailed level. The nature of smart card and the advantages of using smart card are discussed. We also introduced a new kind of smart card – the Java Card. It is a Java based smart card such that we can use Java to program the card. We have employed the Java Card in our proposed system which is discussed in Chapter 5.

Our proposed system is a *Smart Card Certificate* System using *Elliptic Curve Cryptosystem* as the public key algorithm. The main advantage of using elliptic curve cryptosystem is its low computation power requirement. It is most suitable to be used in smart card. Based on the ECC, various kinds of security features can be developed, such as multiple certificates system. To store certificates in a portable device such as

smart card is the most convenience way in the e-commerce world. Yet due to the limitation of computing power and memory of smart card, it seems a little bit difficult to implement it. Our proposed system solves this problem by using ECC. Besides, we also use the "Hyper-Link" concept to store multiple certificates on a single smart card. By using this, the security of doing e-commerce activities can be highly increased.

Payment system is another concern of e-commerce. Traditionally, credit card payment is generally used. However, many people are afraid to give their credit number into the web because if it is known by other people, they can use it to go on-line shopping such that the owner of the credit card does not know it until the date when he receives the bill. Besides, it seems there is no privacy for using credit card. Every transaction is recorded by the bank. Recently, there exists another kind of payment system – electronic cash. It is the digital analogy of paper cash. Therefore it can support anonymity. Privacy can be preserved. It is also an off-line payment. We have given an overview of e-cash in Chapter 6.

Most of the previous papers focus on the 6 main properties of e-cash: Independence, Security, Privacy, Off-line, Transferability and Divisibility. In this thesis, we proposed a new concept of e-cash: Recoverable. It means that if an user has lost his e-cash, he can get it back. It is very easy to achieve this if we give up anonymity. But in our proposed system, both anonymity and recoverability can be achieved.

To conclude, e-commerce will continue to grow. It can be seen that smart card will play a more important role in the coming years. At the same time, electronic cash also has great potential to become a major payment method in this information revolution. This thesis gives two contributions to these areas.

Last but not least, we believe that e-commerce will not only limit to developed countries. People living in other developing countries will also have a taste of it. Our

daily life will become more digital, more comfortable, and more convenience.

Appendix

Papers derived from this thesis

[1] Joseph K. Liu, Victor K. Wei, C. Siu, Roy L. Chan, T Choi. "*Multi-Application Smart Card with Elliptic Curve Cryptosystem Certificate*". Accepted for publication at *EUROCON, July, 2001.*

[2] Joseph K. Liu, Victor K. Wei, Sandy H. Wong. "*Recoverable and Untraceable E-cash*". Accepted for publication at *EUROCON, July, 2001.*

Bibliography

[1] SSH Communications Security Ltd., "SSSH – Cryptography A - Z", http://www.ssh.fi/tech/crypto/

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

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

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

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

[6] 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.

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

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

[9] Bruce Schheier. "Applied Cryptography ", second edition, John Wiley & Sons, Inc.

[10] Alfred J. Menezes, "Elliptic Curve Public Key Cryptosystems", Auburn University Kluwer Academic Publishers, Dordrecht/London, 1993.

[11] Certicom, "Elliptic Curve Cryptography", http://www.certicom.com/research.html

[12] RSA Laboratories, "A Cost-Based Security Analysis of Symmetric and Asymmetric Key Lengths", http://www.rsasecurity.com/rsalabs/bulletins/bulletin13.html

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

[14] M. O. Rabin, R. DeMillo, D. Dobkin, A. Jones, and R. Lipton, "Digitalised Signatures," Foundations of Secure Computation, pp.155-158, Academic Press, 1978.

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

[16] W. Diffie and M. E. Hellman, "New Directions in Cryptography," IEEE Transactions on Information Theory, v. IT-22, n.6, Nov 1976, pp.644-654.

[17] SSL, http://www.ssl.com

[18] SET, http://www.set.com

[19] Nokia, "Nokia on the Web", http://www.nokia.com/

[20] WAP Forum, "WAP Forum", http://www.wapforum.com/

[21] Gemplus, "All About Smart Cards – What is a Smart Card", http://www.gemplus.com/basics/what.htm

[22] Gemplus, "Applications", http://www.gemplus.com/app/index.htm

[23] Mondex, "Mondex", http://www.mondex.com

[24] T. ELGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", IEEE Transactions on Information Theory, Vol. It-31, No. 4 July (1985), Pg. 469-481.

[25] Niels Ferguson. "Single Term Off-Line Coins", Proceedings of Eurocrypt 93, pp. 318-328. 1994

[26] Tatsuaki Okamoto and Kazuo Ohta. "Universal electronic cash". In J. Feigenbaum, editor, *Advances in Cryptology – CRYPTO '91*, Lecture Notes in Computer Science, pages 324-337. Springer-Verlag, 1992.

[27] S. Brands. Untraceable off-line cash in wallet with observers. In *Advances in Cryptology – CRYPTO '93*, Lecture Notes in Computer Science, pages 302-318. Springer-Verlag, 1993.

[28] ZDNet News, "Arizona makes e-voting history", http://www.zdnet.com/zdnn/stories/news/0,4586,2457141,00.html

[29] Yawanna.com, "Surveys show U.S. voters divided on e-voting", http://www.yawanna.com/magazine/6/1000/

[30] Edward Bellamy: "Looking Backward," London, W. Foulsham and Col Ltd. 1888.

[31] eCash Technology, "eCash Technology", http://www.digicash.com/

[32] Berry Schoenmakers, "Basic Security of the ecash™ Payment System," Computer Security and Industrial Cryptography: State of the Art and Evolution, ESAT Course, Leuven, Beligium, June 3-7, 1997. LNCS series, Springer-Verlag

Berlin Heidelberg.

[33] Robert T. Plant, "eCommerce Formulation of Strategy", Prentice Hall, 2000.

[34] EPaynews, "ePaynews", http://www.epaynews.com/

[35] Certicom, "Certicom", http://www.certicom.com/

[36] Motorola, "Motorola", http://www.motorola.com/

[37] Grainger, "Grainger.com", http://www.grainger.com/

[38] Hong Kong Special Administrative Region, http://www.info.gov.hk/

[39] ESDLife Limited, http://www.esdlife.com/