Anonymous and Non-Repudiation E-Cash Scheme Based on Partially Blind Signature

By
Hani M. AL-Matari

Supervised By
Prof. Nidal Shilbayeh

Master Thesis

Submitted in Partial Fulfillment of the Requirements for the Master Degree In Computer Science

Department of Computer Science
Faculty of Information Technology
Middle East University
Amman – Jordan

March, 2011

Middle East University

Authorization Statement

I, Hani M. AL-Matari, authorize Middle East University to supply hard cpies and electronic copies of my thesis to libraries, establishments, or bodies and institutions concerned with research and scientific studies upon request, according to the university regulations.

Name: Hani Mohammed Saleh AL-Matari

Signature:

Date: 27/3/2011

Middle East University

Examination Committee Decision

This is to certify that the thesis entitled "Anonymous and non Repudiation E-Cash Scheme Based on Partially Blind Signature" was successfully defended and approved in March / 2011.

Examination Committee Members	Signature
Prof. Nidal F. Shilbyeh Department of Computer Science Middle East University	
Prof. Musbah M. Aqel Department of Computer Information Systems Middle East University	
Dr. Abdelfatah Aref Tamimi Faculty of Science and Information Technology Al-Zaytoonah University of Jordan	

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to Professor Nidal F. Shelbyaih under whose supervision I chose this topic and began the thesis whilst allowing me the room to work in my own way.

I also extend thanks to the members of my thesis committee for taking the time in reviewing and constructively critiquing my work.

I am deeply indebted to Professor Sattar Aboud, who had laid the foundation of my master degree and then encouraged for research work.

For their efforts, assistance and guidance, special thanks are due to Dr. **Huseen Ouied.**

I cannot end without once again thanking my family, on whose constant encouragement and love I have relied throughout the preparation of this work. It is because of their subtle fostering and inspiring example that I am now the academically and professionally qualified person. To each of the above, I extend my deepest appreciation**.**

DEDICATION

This thesis took a great deal of time and energy, often at the expense of being with my loved ones. This thesis is dedicated to them; to my parents and brothers and sister for loving me unconditionally, for understanding and supporting me in pursuit of my career aspirations.

You have been with me every step of the way, through good times and bad. Thank you for all the guidance and support that you have given me, helping me to succeed and instilling in me the confidence that I am capable of doing anything I put my mind to.

I am grateful for the times my parent, my brother Ahmed have invested in assisting me deliver my work on time, and for the example of my father, **Mohammed AL-Matari,** whose unflinching courage, faith and conviction in all facets of life will always inspire me.

I hope that through this work I am able to –in one way or another repay them for all they have sacrificed in order that I become the person that I am today. It is to them that I dedicate this work.

TABLE OF CONTENTS

List of Tables

.

List of Figures

List of Abbreviations

Term	Definition
CA	Certificate Authority
PKCS	Public Key Cryptography Standard
RSA	Public key cryptosystem Algorithms developed by Rivest, Shamir and Adelman
d	Decryption key
e	Encryption key
SHA	Secure Hash Algorithm.
E-payment	Electronic Payment
E-cash	Electronic Cash
ID	Customer real identifier
PID	Customer pseudo identifier
BO	Blind Office
$\mathbf{S_{cust}}$	Customer signature
$\mathbf{V_{cust}}$	Customer verification
$\boldsymbol{E_{BO}}$	Blind Office's Encryption function
$\boldsymbol{D_{BO}}$	Blind Office's Decryption function
BF	Blind Function
$\mathbf{S_{bank}}$	Bank signature
$\mathbf{V_{bank}}$	Bank verification
v	Predefine message between customer and bank
x	secret signature key
y	public signature key

List of Terminologies

Term	Definition
Asymmetric Cryptography	Is a form of cryptography in which a user has a pair of cryptographic keys; public and private key. The private key is kept secret, while the public key may be widely distributed.
Data Integrity	Assures document authenticity; any change made to the contents of the document will make the signature invalid.
Digest	Used in the process of creating a digital signature, a digest is a unique digital representation or finger print of the signed data.
Digital Certificate	A digital certificate is a computer file which contents three pieces of information: the name(and perhaps another information) of the person or organization to whom it refers, a public (cryptographic) key associated with the person or organization, and digital signature of trusted organization which links the two together .
Digital Signature	Sometimes digital signature referred to as an advance e-signatures take the concept of the traditional paper-based signature into the digital realm, by adding a digital fingerprint as a signature to a document. This fingerprint is unique to both a document and signer.
Encryption	The process of transforming ordinary text data into an

unintelligible form ciphertext. So the original data cannot be either recovered directly that is one-way encryption or through an inverse decryption process that is two-way encryption.

Entity An individual organization, device or process. Used interchangeability with participant.

Hash Function A function that maps a bit string of arbitrary length to fixed length bit string. Approved hash functions are specified in fips 180-2 and are designed to satisfy the following proprieties

1. One–way: it is computationally infeasible to find any input that maps to any new pre-specified output.
2. Collision resistance: it is computationally infeasible to find any two distance input that maps to the same output.

Key A parameter used in conjunction with cryptography algorithm that determines its operation in such a way that an entity with knowledge of the key can reproduce the operation. While an entity without knowledge of the key cannot. Examples applicable to this standard include

1. The computation of digital signature from data.
2. The verification of a digital signature.

Key Generation The trustworthy process of creating private and public key pair.

The public key is supplied to a certificate authority during the

	certificate application process while the private key is only supplied to the subscriber.
Non- repudiation	A service that is used to provide assurance of the integrity and origin of data in such a way that integrity and origin can be verified and validated by third party as having originated from specific entity in possession of the private key that is signatory
Party	An individual organization, device or process. Used interchangeably with entity.
Prime Number Generation	A string of random bits that is used to determine a prime number with the required characteristics.
Private Key	Is an encryption / decryption key known only to the party or parties that exchange secret message. In traditional secret key cryptography, a key would be shared by the communicators so that each could encrypt and decrypt message.
Public Key	Is a value provided by some designated authority as an encryption key that combined with private key derived from the public key, can be used to effectively encrypt message and digital signature.
Security	Features and procedures used to reduce the possibility of fraudulent use, asset compromise, smart card counterfeiting or other

	subversion.
Signature Generation	The process of using digital signature algorithm and private key to generate a digital signature on data.
Signature validation	The mathematical verification of the digital signature and obtaining the appropriate assurance such a public key validity, private key possessions, etc…
Signature Verification	The process of using digital signature algorithm and private key to verify a digital signature on data.
Signed data	The data or message in which a digital signature has been computed.
Smart Card	A card, typically, has the same size a credit card that containts a built-in microprocessor and memory. Smart card used to store user's private keys and in some cases, also perform the hashing value.
Verifier	The entity that verifies the authenticity of a digital signature using the public key.

Mathematical Symbols

Term	Definition
a mod n	The unique reminder r, $0 \leq r \leq (n-1)$,when integer a is divided by n. for example 23 mod 7=2
b ≡ a mod n	There exists an integer k such that b-a = kn; equivalently, a mod n = b mod n. for example, 5 mod 10 = 15 mod 10 = 5.
g	G is generator the q –order cyclic group of $GF(p)^*$; that is, an element of order q in the multiplicative group of GF(p).
gcd (a ,b)	Greatest Common divisor of the integers a and b.
Hash (m)	The result of hash computation (message digest or hash value) on message m using approved hash function.
l	The length of the parameter p in bits.
M	The message that is signed using the digital signature algorithm.
N	The modulus; the bit length of n is considered to be the key size.
p	A prime factor of the modulus n.
q	A prime factor of the modulus n.

Abstract

The blind signature technique Considered since its inception in 1982, as a key principle in building the e-cash schemes until 1996, when emerged the concept of partially blind signature, which played an important role in building electronic cash systems: It allows the signer to include pre-agreed information such as expiration date or collateral conditions in the resulting signature. In this thesis, we proposed a "non-repudiation and anonymous e-cash scheme based on partially blind signature". The main motivation of electronic cash research is providing anonymity. In fact, the unconditional anonymity may be misused for untraceable blackmailing of customers, which is called perfect crime. Furthermore, unconditional anonymity makes ease money laundering, illegal purchase, and bank robbery. Our proposed scheme enables the judge to specify a dishonest customer, bank, or blind office. In addition to that, our scheme is considered as a multi-purpose scheme because it satisfies the integrity and separation of power. We also analyze the efficiency and the security of the proposed scheme.

Multiplicative inverse is a crucial operation in public key cryptography which is used to generate blind signature or partially blind signature. Multiplicative inverse has been widely used in cryptography. Public key cryptography has given rise to such a need, in which we need to generate a related public and private pair of numbers, each of which is the inverse of the other. The basic method to find multiplicative inverses is Extended-Euclidean method. In this thesis we will propose a new algorithm for computing the inverse, based on continuous subtract fraction from integer and divide by fraction to obtain integer that will be used to compute the inverse d. The proposed method is more efficient and faster than the existing methods.

Keywords - Multiplicative inverse, greater common divisor, Euclidean method, Stein method, Gordon method, Baghdad method, e-cash scheme, blind signature scheme, partially blind signature scheme, hash function, RSA scheme, Elgamal scheme.

الـمـلـخص

يعتبر التوقيع الاعمى منذ ظهوره في العام 1982 هو المبدأ الرئيسي لبناء نظم النقد الالكتروني حتى العام 1996م، عندما ظهر مفهوم ما يسمى التوقيع الاعمى جزئيا والذي يلعب الان دورا هاما في بناء انظمة النقد الاكتروني،حيث انـــه يـــسمح للموقع "البنك" ان يضمن معلومات متفق عليها مسبقا مع العميل مثل تاريخ انتهاء صلاحية النقد الالكتروني او اية شـــروط اخرى.

قمنا في هذه الرسالة باقتراح مخطط جديد للنقد الالكتروني اسميناه "مخطط النقد الالكتروني غير قابل للانكـــار والمحـــافظ على الخصوصية المعتمد على التوقيع الاعمى الجزئي " تعتبر خصوصية العميل من اهم دوافع البحث في هذا المجال. في الواقع قد يساء استخدام الخصوصيه لعمل الجريمه الكامله حيث ان الخصوصية الكامله غير االمشروطه قد تؤدي وبسهوله الى عملية غسيل الاموال،الاتجار غير القانوني،او الى السطو على اموال البنك.

يستطيع القاضي من خلال مخططنا وبكل سهوله تحديد الجهة المسؤله عن التلاعب ان وجد سواءً كانت هذه الجهة تتمثل في العميل او البنك او المكاتب العمياء " التي تمثل في مخططنا الجهة الثالثه الموثقه".

يعتبر المخطط المقترح متوافقاً مع سلامة محتوى النقد وكذلك محصناً ضد انتهاك الخصوصية عن طريق جهة واحدة فقط مثلا البنك او المكاتب العمياء كلا على حدة.

ولقد قمنا في هذه الرساله بتحليل الكفاءه والحمايه للمخطط المقترح.

عملية ايجاد المضروب العكسي من العمليات الهامه والمؤثره في تشفير المفتاح العام الذي يستخدم لتوليد التوقيع الاعمى و التوقيع الاعمى جزئيا، ويستخدم المضروب العكسي على نطاق واسع في عملية التشفير.ولقد ساهم تشفير المفتاح العام في اطلاق هذا الاحتياج، حيث اننا عند توليد ارقام عامه وخاصه بحاجه لوجود علاقه بينها بحيث يعتبر كل واحد منها معكوساً للاخر.

تعتبر طريقة اقليديان هي الطريقه الاساسيه لحساب المضروب العكسي.

في هذه الرساله اقترحنا طريقه جديده لحساب المعكوس، تعتمد طريقتنا على الاستمرار في طرح عدد عكسي مـــن اخـــر صحيح ثم نقسمهم على عدد كسري اخر للحصول على عدد صحيح وهذا العدد الصحيح يمثل المعكوس d .

الطريقه المقترحه ذات كفاءه اعلى واسرع من بقية الطرق الموجوده.

Chapter 1

Introduction

Cryptography is an ancient art (kahn, 1996), historically used for secure communication. Cryptography is used of mathematical techniques related to aspects to information security such as confidentiality, data integrity, entity authentication, and data origin authentication (Menezes, et al. 1996). The fundamental object of cryptography is to enable two people to communicate over an insecure channel in such a way that an opponent cannot understand what is being said. This channel could be telephone line or computer network.

Cryptography is the automated (or algorithmic) method in which security goals are accomplished.

Typically, when we say "crypto algorithm" we are discussing an algorithm meant to be executed on computer. These algorithms operate on message in the form of groups of bits.

First we will show the meaning of science of cryptography, which is the study of methods for sending messages in distinct form (namely, in enciphered or disguised form) so that only the intended recipient can remove the disguise and read the message (or decipher it).

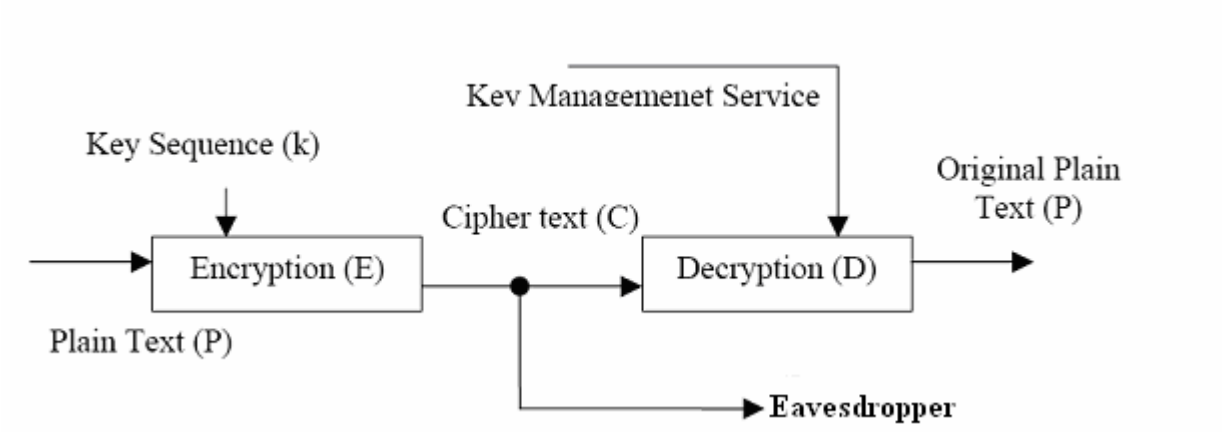


Figure 1.1: Encryption Decryption System

The original message is called the plaintext, and the disguised message is called the ciphertext. The final message, encapsulated and sent, is called a cryptogram. The process of disguising data (binary) in order to hide its information content is called encryption (see figure 1). The reverse process of turning ciphertext into plaintext, which is accomplished by recipient who has the knowledge to remove the disguise, is called decryption or deciphering. Anyone who engages in cryptography is called cryptographer. On the other hand, the study of mathematical techniques for attempting to defeat cryptographic method is called cryptanalysts.

The term cryptology is used to embody the study of both cryptography and cryptanalysts, and the practitioners of cryptology are cryptologists.

The etymology of cryptology is the Greek; kryptos meaning hidden and logos meaning word. Also, the term cipher is a method for enciphering and deciphering.

In the information age, cryptography has become one of the major methods for protection in all the applications.

Cryptography allows people to carry over confidence found in the physical world to the electronic world. Thus it allows people to do business electronically without worries of deceit and deception.

Cryptography in past time was used to assure only secrecy, it was typically used to assure integrity of the message and authenticity of the sender.

When people started doing business online and needed to transfer funds electronically, the applications of cryptography for integrity began to surpass its use secrecy. Hundreds of thousands of people interact electronically every day, whether it is through e-mail, e-commerce (business conducted over the internet), ATM machines, or cellular phones. The constant increase of information transmitted electronically has led to an increased reliance on cryptography. During and before World War II, the main applications of cryptography were military. Both coding theory and cryptography originated with the seminal work of Claude Shannon in 1948.

After the war, with computer spread and electronic communication, the cryptography schemes were used for password, banking transaction and various aspects of computer security proliferated.

An obvious application of cryptography is transformation of information to prevent others from observing its meaning; this is the classical concept of secrecy. Secure communication is the most straightforward use of cryptography.

1.1 Electronic cash scheme

Electronic money (also known as e-currency, e-money, electronic cash, digital money, digital cash or digital currency) is a technology that uses varied electronic methods to execute financial transaction. Typically, this involves the use of computer network, the internet and digital stored value system. Electronic funds transfer (EFT) and direct deposits are all examples of electronic money. Physical cash is token-based fiat money, where the value token are coins or bank notes. The token are produced in such a way that it is easy to verify them as being genuine. Perhaps through a watermark or metal strip in a bank note, but they are very difficult to forge. The token represents the actual monetary value and transfer of them completes a payment, without any transaction fees. If the current holder destroys the tokens then it has lost the value they represent. Cash allows payer anonymity and payment untracebility, since there is no information on the token to link the payment to the payer, unlike payment card and cheque transaction. A bank could conceivably record the unique serial number on a bank note when it is withdrawn, in the hope to discover who later deposits the note. However due the transferability of cash, in that it can be passed from person to person in payment indefinitely, without being returned to the bank, it is impossible to trace its path without cooperation of each party it passed through. Coins do not have such serial numbers and cannot be traced. The history of money shows that the ways of representing

value have become increasingly abstract over time (Davies, 1996). The term electronic cash is often applied to any electronic payment system that appears to offer any of the above attributes of physical cash. The good way to think electronic cash as a direct electronic macro payment system, where the payment instrument consists of prepaid payee-independent electronic value issued by a trusted financial agent. This definition excludes account-based systems that rely on the authenticated transfer of value between accounts. Unlike electronic value token, if the account transfer message is destroyed, value is not "lost" by the payer. The definition also excludes bank-certified electronic cheque, which are vendor dependent, in that the payee is identified in the cheque before payment occurs. As money has evolved so have the methods of effecting payment. Modern payment systems are either account based or token based. Payment instrument, which effects transfer between accounts, includes cheques, payment cards (credit or debit based) and bank giros. Token based system includes cash, pre-paid phone cards, and postal stamps. An electronic payment system or network payment system is based on existing payment instruments while others introduce a new form of value representation and exchange. An electronic cash scheme consists of three activities; withdrawal, payment, and deposit phases. A customer withdraws electronic coins from her bank. A coin consists of some data, to uniquely identify it. Each coin is digitally signed by the bank to show that it is authentic. To make a payment, the customer sends the coins across network to payee. The payee can verify the bank's signature on coin. To ensure that these coins have not been spent before, the payee deposits the coin in the bank, for verification. The bank maintains a database of all spent coins, to prevent double spending of tokens. If the coins have already been spent then this coin's data will be present in the data base. Otherwise, the payment is valid, and the coin's data is then entered into the database, having now been spent.

The previous scheme lacks some of the properties of physical cash such as anonymity and

transferability. The bank must be contacted on line during each purchase to prevent double spending.

1.2 Problem Definition

Electronic commerce and electronic business greatly need new payment scheme that will support their further development. As we mentioned, the basic e-cash scheme lacks some of the properties of the physical cash such as anonymity and untraceability. Recently, much work has been done to extend and improve the basic scheme. Multiplicative inverse is a crucial operation in public key cryptography which is used to generate blind signature and partially blind signature. The modular inverse problem is difficult to solve. Sometimes it has a solution sometimes not.

In general, a^{-1} a x (mod n) has a unique solution if a and n are relatively prime. If a and n are not relatively prime, then a^{-1} a x (mod n) has no solution. If n is a prime number, then every number from 1 to n-1 is relatively prim to n and has exactly one inverse modulo n in that range. Because of the difficulty to solve the inverse problem, (Schneier, 1996), only a couple of methods for compute the inverse.

1.3 Objectives of the Study

- We developed a new method for computing the multiplicative inverses.
    i. The proposed method has the following advantages:
        - ❖ Simple.
        - ❖ Need less storage.
        - ❖ Approximately irrelevant to e or n
    ii. Our method has been proved and compared with the existing methods.
- We developed an efficient e-cash scheme based on RSA partially blind signature.
    i. The proposed scheme has the following advantages:

- ❖ It is one of the first schemes that achieves prevention of any type of extortion threats? with the partially blind signature.
- ❖ Preserves all blind signatures' feature and takes all advantages in partially blind signature.

ii. Scheme's correctness has been proved

iii. Scheme's security has been proved.

1.4 Thesis significance

The significance of the study lies in the following:

Introduce new algorithm for computing multiplicative inverse which is a crucial operation in public key cryptography, and has been widely used in cryptography.

Introduce new e-cash scheme that preserves customer's anonyms and enables the judge to specify the dishonest entity (customer, bank or blind office). Also protect bank's data base from growing unlimitedly via using the partially blind signature.

1.5 Thesis Organization

This thesis is organized as follows: chapter 1 is the introduction chapter. Chapter 2 describes literature review for e-cash scheme and multiplicative inverse and illustrates the properties of e-cash scheme. Chapter 3 describes the previous methods and proposed Fast Fraction Integer Method (FFIM) for computing multiplicative inverse and illustrates the correctness of the (FFIM) proposed method. In Chapter 4 we described the proposed e-cash scheme and illustrated the correctness of the proposed scheme. Also this chapter contains the security of the proposed e-cash scheme. Finally we state concluding remarks and future work in chapter 5.

Chapter 2

Literature Review

2.1 E-cash Scheme

Digital cash or e-cash scheme is a set of parties with their interaction, exchanging money and goods. A typical e-cash system has three parties: as shown in figure 2.1

Customer: purchases goods or services from payee using e-cash.

Bank: issues e-cash and maintains bank account for customer and payee.

Payee: sells goods or services to customer, and deposit e-cash to bank.

And there are also three activities, withdrawal, spending and deposit.

A customer withdraws electronic coins from bank and pays the coins to a payee. Finally, the payee deposits the paid coins to the bank.

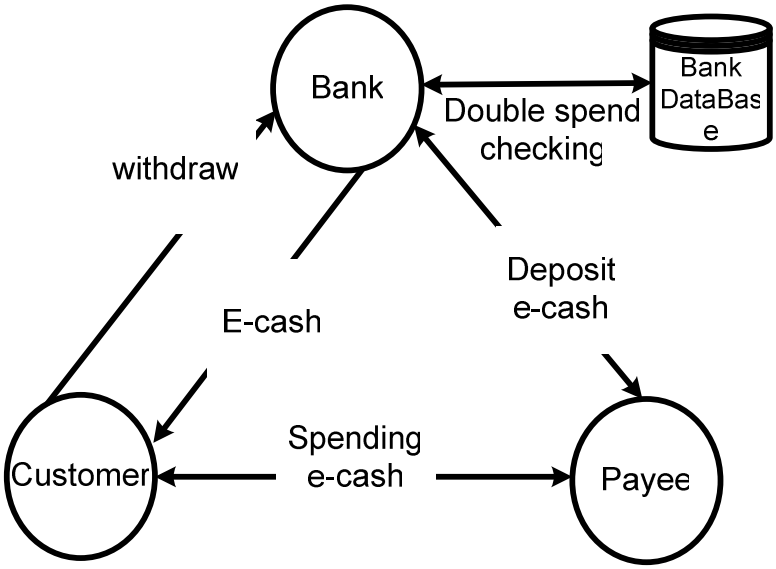


Figure 2.1: Basic model of e-cash system

The concept of electronic cash was proposed by (Chaum, 1992). The main idea is that, a bank is responsible for giving out electronic coins, and for later accepting them for deposit, the withdrawal and spending protocols are designed in such a way that it is impossible to identify

when particular coin was spent. i.e. the withdrawal protocol does not reveal any information to the bank that would later enable others to trace how a coin was spent. Chaum's scheme is based on RSA public key cryptosystem and its security depends on the difficulty of integer factorization. Blind signature schemes see a great deal of use in application where customer privacy is important.

In the beginning, it is better to have a complete image about how e-cash must explain the blind signature scheme.

2.1.1 Blind signature scheme

The basic idea is that sender A sends a message m to signer which signer signs and return to sender A. according to this signature, sender A can compute signer's signature on a message m, the signer knows neither the message in nor the signature associated with it i.e. the signer is unable to associate the signed message with the sender A.

This may be important in electronic cash applications where a message m might represent a monetary value that sender A can spend. When m and bank signature S_B (m) are presented to bank for payment, bank is unable to deduce which party was originally given the signed value. This allows the customer to remain anonymous.

The blind signature scheme requires the following components

A digital signature scheme for the singer B. $S_B(x)$ denotes the signature of Bank on x.

Function f and g (known only to sender (customer)) so that g ($S_B(f(m))$) = $S_B(m)$. f is called a blinding function, g an unblinding function, and f(m) a blinded message.

2.1.1.1 Blind RSA signature Scheme

One of the simplest blind signature schemes is based on RSA signing (Rivest, et al. 1978) as shown in figure 2.2.

- **Initializing:**

The bank randomly chooses two large prime number p and q, and computes $n = p \cdot q$ and φ $(n) = (p\text{-}1)(q\text{-}1)$. It then determines a pair of public and private keys(e,d), satisfying $e \cdot d \equiv 1 \pmod{\varphi(n)}$ with $gcd(e, \varphi(n)) = 1$ (see section 3.3), and both e and d less than φ (n). The bank publishes (e, n).

- **Withdrawing:**

If customer decides to withdraw e-cash from the bank, he/she randomly chooses two integer m and r in Z^*_n, such that r is relatively prime to n (i.g. gcd(r,n)=1). r is raised to the public exponent e modulo n, and the resulting value re mod n is used as a blinding factor. The author of message (customer) computes the product of the message and blinding factor (i.e. $m' \equiv mr^e \pmod{n}$ and sends the resulting value m' to signing authority (Bank). Because r is random value and mapping $r \rightarrow r^e \bmod n$ is permutation it follows that r^e mod n is random too. This implies that m' does not leak any information about m, the signing authority (Bank) then calculates the blinded signature s' as: $s' \equiv (m')^d \pmod{n}$.

s' is sent back to author of message (Customer).

- **Unblinding**:

After receiving the s' the author of message (customer) can then remove the blinding factor to reveal s, the valid RSA signature of m: $s \equiv s' \cdot r^{-1} \pmod{n}$.

This works because RSA keys satisfy the equation: $r^{ed} \equiv r \pmod{n}$ and thus

$s \equiv s' \cdot r^{-1} \equiv (m')^d r^{-1} \equiv m^d r^{ed} r^{-1} \equiv m^d r r^{-1} \equiv m^d \pmod{n}$ hence s is indeed the signature of m.

- **Depositing:**

When the customer uses the e-cash, the payee verifies that $s^e \equiv m \bmod n$. if they are correct, he\she calls the bank to check whether the e-cash has been already spent, i.e. double spending checking. If the e-cash has not been spent, the payee accepts the payment and deposits the e-cash into his/her account, and the bank stores (m,s) in its database for double-spending checking, and adds coins to the payee's account.

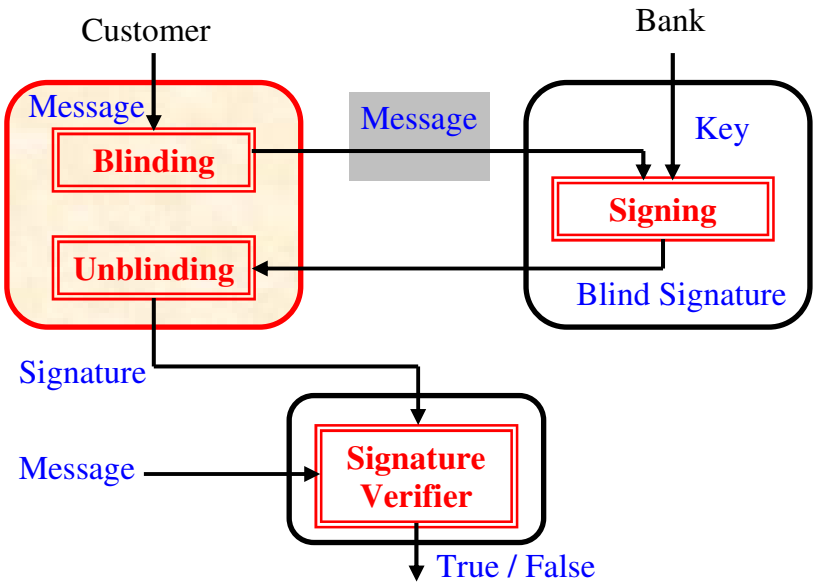


Figure 2.2: Blinding signature.

2.1.2 Partially Blind Signature Scheme

The notion of partially blind signature was first introduced by (Abe and Fujisaki, 1996). It was construction, based on RSA. In a partially blind signature scheme the signature can impose the common information, such as the value and expiry date. The common information and the signature can check the validity of this signature (Abe and Camenisch, 1997). In their scheme, the bank is clearly notified the common information such as expiration date of an e-cash. With the partially blind signature, the bank assures that the signed e-cash carry the

agreed common information. With this common information such expiration date, the bank needs only to keep the still-alive e-cash in the database to prevent double-spending. Those expired e-cashes could be eliminated from database without any trouble. This partial blindness property preserves the unlinkability of the blind signature, but imposes the common information on the signature.

Based on the discrete logarithm problem, (Miyazaki, et al.,1997) proposed a partially blind signature, and proposed an efficient E-cash system.

Based on Quadratic Residue (QR) theory, Fan and Lei (1998) proposed the partially blind signature scheme, and there is no modular exponentiation and inverse computations performed by the signature requesters.

Juang and Lei(1999), based on the discrete logarithm problem, proposed a partially blind (*t, n*) threshold signature scheme in which any *t* out of *n* signers in a group can represent the group to sign the partially blind threshold signature.

(Hwang, et al. 2002) showed that Fan-Lei's (1998) scheme could not meet the untraceability property of a blind signature.

Huang and Chang (2004) proposed a new design of efficient partially blind signature based on discrete logarithm and the Chinese Remainder.

Zhang and Chen (2005) show that Huang and Chang partially blind signature scheme is not secure.

2.1.2.1 Review of Abe and Fujisaki's Scheme

Abe and Fujisaki's partial blind signature scheme as shown in figure 2.3. designed to protect the bank's database from growing without limits since the bank needs to store spent e-cash in its database for double-spending checking. In the scheme, each e-cash issued by the bank contains an expiration date so that all expired e-cash recorded in the bank's database can be

removed, (Abe and Fujisaki, 1996). The partial blind signature scheme is described as follows:

- **Initializing:**

Based on RSA public key cryptosystem (Ravest, et al., 1976), the bank randomly chooses two large prime number p and q, and computes $n = p \cdot q$ and $\varphi(n) = (p-1)(q-1)$. It then determines a pair of public and private keys(e,d), satisfying $e \cdot d \equiv 1 \pmod{\varphi(n)}$ with $\gcd(e, \varphi(n)) = 1$ (see section 3.3), and both e and d less than $\varphi(n)$. the bank publishes (n, e, f), which f is an appropriate public exponent generation function, f(v) must be different for different value of v; where v is predefine message that contains expiration date of the e-cash.

- **Withdrawing:**

If a customer decides to withdraw e-cash from the bank, he/she randomly chooses two integers m and r in Z^*_n, where m is a message and r is a blind factor; r is relatively prim to n and computes $\alpha \equiv (r^{ev}. m \bmod n)$ where v is a message predefined by the bank and contains an expiration date of the e-cash. The customer then sends α and v to the bank. After receiving (α, v), the bank first verifies whether or not v is correct. If it is correct, the bank sends $s' \equiv (r^{(ev)-1} \bmod n)$ to the customer and deducts true coins from customer's account in the bank.

- **Unblinding:**

After receiving s', the customer computes $s \equiv (r^{-1}s' \bmod n)$ and gets his/her e-cash (m, s, v).

- **Depositing:**

When the customer uses the e-cash, the payee first verifies v is correct and $s^{ev} \equiv$ m mod n. If they are correct, he/she calls the bank to check whether the e-cash has been already spent, i.e. double spending checking. If the e-cash has not been spent, the payee accepts the payment and deposits the e-cash into his/her account, and bank stores (m, s, v) in its database for double-spending and adds money to the payee's account.

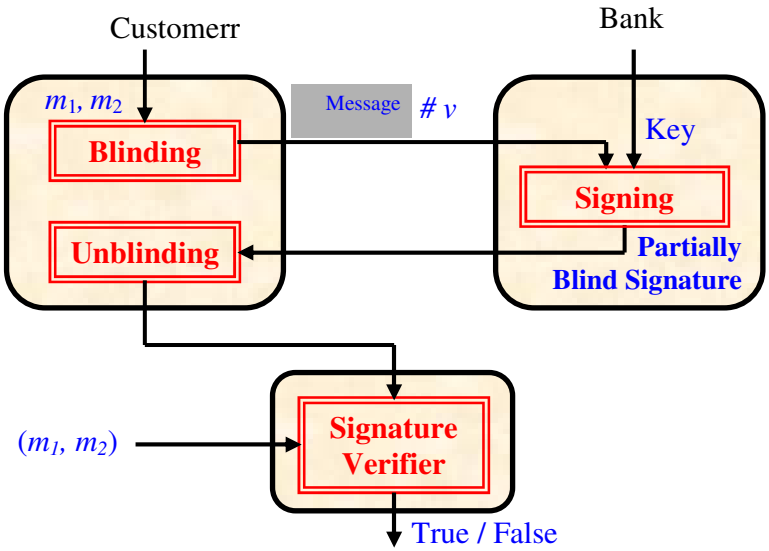


Figure 2.3: Partially blinding signature.

2.1.3 Anonymity and Untraceability

The main motivation of electronic cash research is providing anonymity (Peirce, 2000). Different levels of payer anonymity are possible with different electronic payment systems.

Anonymity related requirement:

Payer anonymity: the identity of the payer should remain unknown; the payer must not be identifiable.

Payer untraceability: inability to trace a payment back to payer.

Payment untraceability: inability to trace a payer to obtain information about the payments he/she has performed.

Payment unlinkability: inability to link different payments, as made by the same payer. The payer anonymity must be achieved towards the payee, bank, and any other party involved in designed system. In some systems payer anonymity, payer untraceability and payment untraceability are provided towards the payee, but not payment unlinkability.

In fact the anonymity might be misused by fraudulent to perform a perfect crime (Solms and Naccache,1992). However in Solms and Naccache (1992) have shown that unconditional anonymity may be misused for untraceable blackmailing of customers, which also called perfect crime. Furthermore, unconditional anonymity makes ease money laundering, illegal purchase, and bank robbery. Such as stealing of the private keys, money laundry, and blackmailing of coins. The use of blindfolded protocols in the bank is considered as modern threats (Liu, et al., 2001). These threats are just prevented in the scheme of (Hohenberger, 2006), which is also not efficient as it needs the trusted authority interaction in e-payment schemes.

Interesting issue is the revocability of the anonymity (Peirce, 2000). Electronic payments are an interactive field for fraud. Implementing mechanisms to protect the anonymity of honest customer should not give facilities to malicious customer to carry out fraud while their identity is protected. Therefore, we need mechanisms to prevent fraud or, at least, to be able to detect it and trace the dishonest entity. This should be achieved without affecting the anonymity of honest customer. It is understood that the trustee(s) is the only entity that can perform revocation, and will only answer a request if there exists sufficient evidence that a transaction is not lawful (Camenish, et al., 1996),(Claessens, et al., 1999).

Anonymity in e-cash must be controlled by some requirement as a follows (Davida, et al., 1997):

- ***Anonymity:*** for legitimate customer; the following anonymity requirements should be provided for honest customer.
    - ✓ Payer anonymity.
    - ✓ Payer untraceability.
    - ✓ Payment untraceability.
    - ✓ Payment unlinkability.
- ***Revocation:*** to prevent fraud, anonymity should be revocable in certain cases, for example with a judge's order. In this case, a trusted party, or a combination of parties should be able to perform owner tracing or coin tracing.
- ***Separation* of power:** the trustee(s) which have the ability to revoke anonymity should not have any power other than tracing; in particular, they should not able to forge coins or impersonate customer.
- ***No framing*:** even if the bank collaboration with the trustee(s) or other parties (payee, etc), should not be able to impersonate customer.
- ***Selectivity***: revocation must be selective; that is, only the transaction for which judge's order is given must be de-anonymized. The system must behave as a fully traceable system with respect to this transaction, but anonymous for the rest even for transaction of the same customer.
- ***Efficiency*:** the added burden for the basic system should be minimal for all involved parties. In particular, trustee(s) must be involved only when revocation is required, and remains off-line otherwise.
- ***Crime prevention***: anonymity revocation should not (even indirectly) motivate crimes more serious that those it protects against. The blind signature technique makes anonymous electronic payments possible as follows: the customer requests the signer to sign on blinded data. The customer then derives the wanted signature from the signed blinded data. When

the customer finally hands in message and its signature, the signer is able to verify this signature, but is unable to link this signed message to previous signing process instance.

2.1.4 Online and off-line issue

The online and off-line issues in electronic cash scheme refer to a specific property of the spending protocol. The spending protocol is functionally protocol between two parties (customer and payee). If payment systems require that the payee contact a third party (the bank) before accepting a payment, the system is called online payment system; the communication may be any medium (no necessary the internet). In off-line system the contact with third party (the bank) is not required during the spending protocol.

The coins are represented by data, so that it is easy to duplicate data, an electronic cash scheme requires a mechanism that prevents a customer from spending the same coin twice (prevents double spending) or at lest traces double spending. There are two scenarios; in the on-line scenario (Chaum, 1983, 1985, 1989), the bank is online in each transaction to ensure that no coin is spent twice, and the payee must consult the bank before accepting a payment. In offline scenario (Chaum, et al., 1988), a payee accepts the payment autonomously and later submits the payment to the bank; the payee is guaranteed that such a payment will be either honored by the bank, or will lead to trace the double spending (and therefore punishment) of the double-spender.

Two trace mechanism that can be used to provide anonymity control (Davida, et al., 1997):

- ✓ ***Owner tracing****:* allows the trustee(s) to disclose the identity of the owner of specific When a coin has been double spent or used in some other fraudulent way; with an owner tracing it would be possible to find out the owner of the coin. The protocol can only be applied after the fact, that is, the crime has been committed.

- ✓ ***Coin tracing:*** traces the coin(s) originated from suspicious withdrawal. With this mechanism trustees can trace the destination of the coins. This might be useful to prevent blackmailing, illegal good selling, etc. with the coin tracing trustees can trace activities of suspect party, and trace coins before they have been spent.

2.1.5 Review of Some previous Schemes

There are many schemes that have solved the anonymity problem. In this section, we are going to reviews some schemes.

❖ CAFÉ

CAFÉ (Boly et al. 1994) is an off-line payment system. It is based on blind signature technique, but here the coin contains information about the identity of the payer, that is only revealed in case of double spending. The system lies in the customer's responsibility for its own anonymity. Under normal circumstance (honest customer), the system provides customer anonymity, payment untraceability, customer (payer) untraceability and payment unlinkability. In this scheme if the customer does not use anonymous connections an eavesdropper, the payee and the bank may disclose his identity or link together withdrawal and payment or different payments.

NetCash

In this system (Medvinsky and Neuman, 1993) there is a framework that supports realtime electronic payments with prevision of anonymity over an unsecure network.

There are three parties: currency service, merchant (including bank and payees) and customer. The Currency service knows the serial number of the coins and the identity of the

customer during the withdrawal phase. The system provides payment unlinkability, customer anonymity and payment untraceability toward the payee and the bank.

.

❖ Gemplus model

In (M'riahi and Pointcheval, 1998), the system is based on blind signature. In the model blinding is sub-contracted to trustee, using identity linked pseudonyms (PIDs) to achieve anonymity. The bank acts as certification authority that provides the customer the certified PIDs, therefore the bank links the pseudonyms and real identity of the customer. The trustee is designed as blind office BO, it can link all payments made under the same PID. The customer uses the PIDs to pseudo-identity himself to the BO. Anonymity is achieved by transferring the capability to link certain coin and the customer ID from bank to BO. The implemented anonymity is revocable by collaboration entities (bank and BO), therefore the privacy of the customer is vulnerable towards collaborating entities.

2.2 Multiplicative Inverse

Modular arithmetic plays an important role in cryptography. Many public-key schemes (Gordon, 1989) involve modular exponentiation. The multiplicative inverse of number is simply a value that results in one when you multiply it by the original number. In the world non-modular arithmetic, finding the multiplicative inverse of an integer is trivial; reciprocal. You need hardly to prove that, for all integer x not equal to zero, the following holds: $x * \frac{1}{x} = 1$. While $a^{-1} \equiv x \pmod{n}$ is significantly harder than in non-modular case. For one thing, not all numbers will have an inverse modulo another number.

Generally, the equation $a^{-1} \equiv x \pmod{n}$ has a unique solution only if a and n relatively prime, i.e gcd(a, n)=1(Okamoto, 1994).

Based on RSA cryptosystem and in key generator stage to compute the decryption key *d,* we can use one of the algorithms as in figure 2.4. The keys for the RSA algorithm are generated in the following way:

- Choose two distinct large random prime numbers *p* and *q*.
- Compute *n*= *pq*.(n is used as the modulus for both the public and private keys).
- Compute *Ø(n) = (p - 1) (q - 1).*
- Choose an integer *e* such that *1*< *e* < *Ø(n)*, and *e* and *Ø(n)* share no factors other than 1 (i.e. *e* and *Ø(n)* are coprime) (*e* is released as the public key exponent).
- Compute *d* to satisfy the congruence relation ed ≡ 1 mod *Ø(n).* (*d* is kept as the private key exponent).

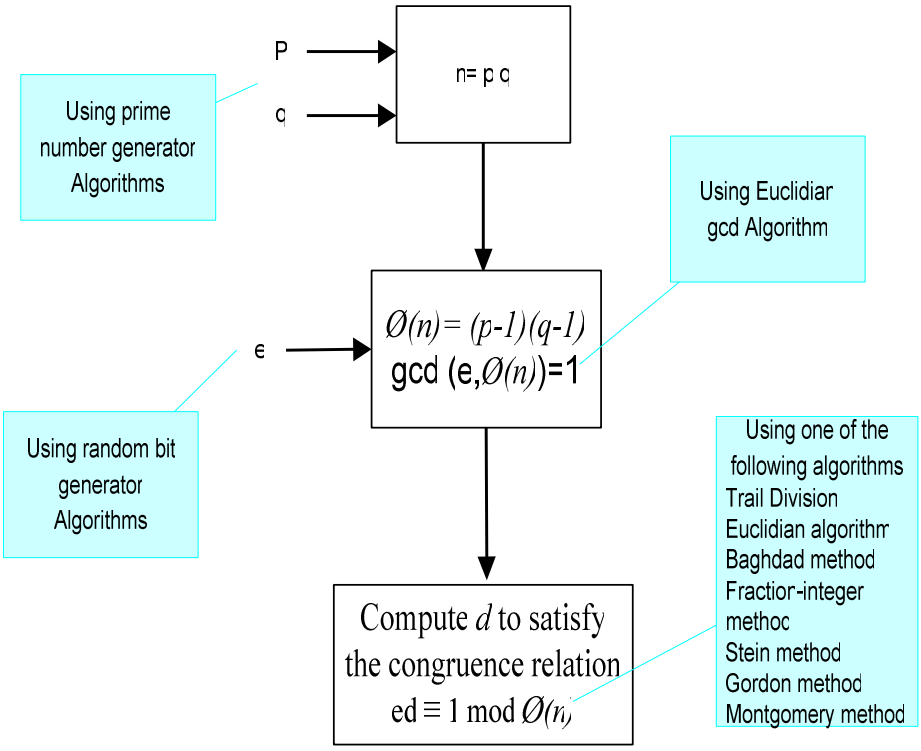


Figure 2.4 RSA key generator stage

The modular inverse problem is too difficult to solve. Sometimes it has a solution, sometimes not. For example, the inverse of 5, modulo 14, is 3. On the other hand, 2 has no inverse modulo 14.

In general, a^{-1} a x (mod n) has a unique solution if a and n are relatively prime. If a and n are not relatively prime, then a^{-1} a x (mod n) has no solution. If n is a prime number, then every number from 1 to n- 1 is relatively prime to n and has exactly one inverse modulo n in that range.

Because the difficulty to solve the inverse problem (Schneier, 1996), only a couple of methods for compute the inverse. The multiplicative inverse of (e) modulus (n) is an integer (d) Z ϵ n such that e.d ≡ 1 mod n, d is called the inverse of e and denoted e^{-1} (Rivest, Shamir, and Adlemen, 1978).

2.2.1 Review existed method

Knuth, D. E. (1981) Euclidean described the algorithm in his book, *Elements*, written around 300 B.C. He didn't invent it. Historians believe the algorithm could be 200 years older. It is the oldest nontrivial algorithm that has survived to the present day, and it is still a good one. Euclidean's algorithm is an efficient way to calculate the greatest common divisor of two integers without factoring them.

Euclidean's algorithm can also compute the inverse of a number modulo *n.* Sometimes this is called the *extended Euclidean algorithm.* This method is based on the idea that if n>a then gcd (a, n) = gcd (a, n mod a), also on finding a.x+y.n=1 in which x is the multiplicative inverse. Euclidean algorithm is approximately irrelevant to e or n, but other algorithms are affected by e and n.

In the algorithm was described by Stein, J. (1967) and improved by Knuth, D. E.(1981), which avoids multiplications. It is based on the observation that

gcd(x,y)=gcd(x/2,y) if x is even, also gcd (x,y) =2gcd (x/2,y/2) if both x, y are even, and gcd(x,y) = gcd((x-y)/2,y) if x, y are both odd.

Gordon in (1989) described the algorithm which was based on the observation that (q) at Euclidean algorithm does not need to be the remainder of n/a but it can be any power of 2 up to that limit.

The algorithm which was described by Sattar J. (2004) is based on the Baghdad method it is very simple, including addition 1 to n and then divides the result by a and keep on adding the result to n and divides the new result by a until an integer is obtained.

In the algorithm was described by Sattar J. (2005).The idea behind the *(FIM)* method is very simple. Start with divide 1 by e, and divide n by e, then keep on adding the two results in any variable until an integer is obtained.

It is lengthy in calculating the inverse, because it is a sequential search. (I.e. Start by d = 1, keep on adding 1 to d until e * d ≡ 1 mod n).

Chapter 3

Fast Fraction-Integer Method for Computing Multiplicative Inverse

3.1 Introduction

Multiplicative inverse is a crucial operation in public key cryptography, and has been widely used in cryptography. Public key cryptography has given rise to such a need, in which we need to generate a related public and private pair of numbers, each of which is the inverse of the other. The basic method to find multiplicative inverses is Extended-Euclidean method. In this thesis we will propose a new algorithm for computing the inverse, based on continuous subtract fraction from integer and divide by fraction to obtain integer that will be used to compute the inverse d. We claim that the proposed method is more efficient and faster than the existed methods.

Modular arithmetic plays an important role in cryptography. Many public-key schemes Gordon, (1989) involve modular exponentiation. Modular inversion, the computation of $b^{-1} \bmod a$ has a part in exponentiation based on addition-subtraction chains, as well as other applications in such public key systems.

The multiplicative inverse of e modulus n is an integer d such that $e*d \equiv 1 \bmod n$, d is called the inverse of e and denoted e^{-1} (Rivest, et al.,1978). The study of inverse calculation was an intractable science due to lack of real improvement, the modulus inverse problem is a lot more difficult to solve (Schneier, 1996). However, there were only a few methods.

The first one is trivial and lengthy in calculating the inverse, because it is a sequential search. It starts by $d = 1$, keep on adding 1 to d until $e*d \equiv 1 \bmod n$.

Euclidean described the algorithm in his book, Elements, written around 300 B.C (Knuth, 1981). It is the oldest nontrivial algorithm that has survived to the present day, and it is still a good one. Euclidean's algorithm is an efficient method to calculate the greatest common divisor of two integers without factoring them.

Euclidean algorithm can also compute the inverse of a number modulo n, sometimes this is called the extended Euclidean algorithm, this method is based on the idea that if $n > a$ then $\gcd(a,n) = \gcd(a, n \bmod a)$, also on finding $a*x + y*n = 1$ in which x is the multiplicative inverse.

Euclidean algorithm is approximately irrelevant to e or n, but other algorithms are affected by e and the modulus n.

3.2 Previous methods

In this section we will describe the methods that deal with the computing multiplicative inverse which are as follows:

3.2.1 Euclidean algorithm

This method is based on the idea that if $n > e$ then gcd(e,n) =1, also on finding $e*x + y*n$ =1 in which x is the multiplicative inverse of e (Menezes et al, 1996). The algorithm is iterative and can be slow for large numbers. Knuth showed that the average number of divisions performed by the algorithm is: 843*$\log_2(n)$ + 1.47 (Knuth, 1981).

Algorithm

Input: $e \in Z_n$ such that gcd (e, n) = 1

Output: e^{-1} mod n, where e^{-1} = i provided that it exists

1 Set g ← n, u ← e, i ← 0, v ← 1;

2. while u > 0 do the following:

2.1 q ← [g/u], t ←g - q * u;

2.2 g ← u, u ← t, t ← i – q * v;

2.3 I ← v;

2.4 v ← t;

3. if i < 0 then i ←n + i;

4. e^{-1} ← i;

Table 3.1 shows an implementation of the Euclidean algorithm.

Let $e \leftarrow 7; n \leftarrow 60$

g	u	i	v	q	t
60	7	0	1	0	0
7	4	1	-8	8	-8
4	3	-8	9	1	9
3	1	9	-17	1	-17
1	0	-17	-52	3	-52

Table 3.1: Result for Euclidean algorithm

$e^{-1} \leftarrow n + i = 60 + (-17) = 43$

The method needs 8 variables, and used subtraction, multiplication, division, and comparison as operations.

3.2.2 Stein Method

In 1967, Stein introduced an inverse algorithm (Stein, 1967) and later improved by (Knuth,1981). It is based on the observation that $\gcd(x, y) = \gcd(x/2, y)$ if x is even, also $\gcd(x, y) = 2$, $\gcd(x/2, y/2)$ if both x, y are even, and $\gcd(x, y) = \gcd((x - y)/2, y)$ if x, y are both odd. The algorithm needs 11 variables, and uses addition, subtraction, multiplication, division and comparison, the complexity is $O(\log n)$.

Algorithm

Input: $e \in Z_n$ such that $\gcd(e, n) = 1$.

Output: $e^{-1} \bmod n$ provided that it exists.

The algorithm is as follows:

While e and n is even do
$e \leftarrow \lfloor e/2 \rfloor, n \leftarrow \lfloor n/2 \rfloor$;
$u_1 \leftarrow 1; u_2 \leftarrow 0; u_3 \leftarrow e; v_1 \leftarrow n; v_2 \leftarrow 1-e; v_3 \leftarrow n;$ If
e is odd then
$t_1 \leftarrow 0; t_2 \leftarrow -1; t_3 \leftarrow -n;$
Else $t_1 \leftarrow 1; t_2 \leftarrow 0; t_3 \leftarrow e;$
Repeat
While t_3 is even do
If t_1 and t_2 is even then
$t_1 \leftarrow \lfloor t_1/2 \rfloor, t_2 \leftarrow \lfloor t_2/2 \rfloor$;
Else $t_1 \leftarrow \lfloor (t_1+n)/2 \rfloor, t_2 \leftarrow \lfloor (t_2-e)/2 \rfloor$;
If $(t_3 > 0)$ then
$u_1 \leftarrow t_1; u_2 \leftarrow t_2; u_3 \leftarrow t_3;$
Else $v_1 \leftarrow n-t_1; v_2 \leftarrow -(e+t_2); v_3 \leftarrow -t_3;$
$t_1 \leftarrow u_1 - v_1; t_2 \leftarrow u_2 - v_2; t_3 \leftarrow u_3 - v_3;$
If $(t_1 < 0)$ then
$t_1 \leftarrow t_1 + n; t_2 \leftarrow t_2 - e;$
Until $t_3 = 0;$
$e^{-1} \leftarrow u_1;$

Table 3.2 shows an implementation of the Stein algorithm.

Let e ← 7, n ← 60.

e	n	u1	u2	u3	v1	v2	v3	t1	t2	t3
7	60	1	0	7	60	-6	60	0	-1	-60
								30	-4	-30
								15	-2	-15
					45	-5	15			
								-44	5	-8
								16	-2	
								8	-1	-4
								34	-4	-2
								17	-2	-1
					43		1			
								-42	5	6
								18	-2	
								9	-1	3
		9	-1	3						
								-43	4	2
								26	-3	
								43	-5	1
		43	-5	1						
								0	0	0

Table 3.2: Result for Stein algorithm

3.2.3 Gordon Method

Gordon (1989) described another algorithm for computing an inverse. It is based on the observation that q at Euclidean method does not need to be the remainder of n/a but it can be any power of 2 up to that limit e (Menezes et al, 1996). The algorithm needs nine variables, and uses addition, subtraction, comparison, and shifting. The complexity of the algorithm is $O(\log n)$

Algorithm

Input: $e \in Z_n$ such that gcd (e, n) = 1
Output: e^{-1} mod n, provided that it exists

1. g ← n, i ← 0, v ← 1, u ← e.
2. repeat
   - 2.1 s ← -1, p ← 0.
   - 2.2 If u > g then
     - 2.2.1 t ← 0
   - 2.3 else
     - 2.3.1 p ← 1, t ← u.
     - 2.3.2 while (t ≤ g) do
       - 2.3.2.1 s ← s + 1.
       - 2.3.2.2 t ← left shift t by 1.
     - 2.3.3 t ← right shift t by 1.
   - 2.4 t ← g - t, g ← u, u ← t, t ← i, i ← v.
   - 2.5 if p =1 then
     - 2.5.1 v ← left shift v by s.
     - 2.5.2 t ← t - v.
   - 2.6 v ← t.
3. until u = 0 or u = g.
4. if i < 0 then i ← n + i.
5. e^{-1} ← i.

Table 3.3 shows an implementation of the Gordon algorithm.

Let $e \leftarrow 7; n \leftarrow 60$

g	u	i	v	s	p	t
60	7	0	1	0	1	14
				1		28
				2		58
				3		112
						56
7	4					4
		1				0
			8			−8
				−1	0	
					1	4
				0		8
4	3					3
		-8				1
						9
			9			
				−1	0	
					1	3
				0		6
						3
1	1					1
		9				−8
						−17
			−17	−1	0	
					1	1
				0		2
				1		4
						2
1						1
		−17				9
			−3			43
			43			

Table 3.3: Result for Gordon algorithm

3.2.4 **Baghdad algorithm**

(Aboud, 2004) introduced another algorithm entitled "Baghdad method" to calculate the inverse. The idea behind Baghdad method is very simple involving adding 1 to the modulus n and then divides the result by the exponent e. Then keep on adding the result to the modulus n and divide the new result by the exponent e until an integer is obtained.

The algorithm needs only 5 variables, and uses addition and division only. The complexity of the algorithm is $O(\log n)$

Algorithm

Input: $e \in Z_n$ such that $\gcd(e,n) = 1$

Output: $e^{-1} \bmod n$ provided that it exists

The algorithm is as follows:

Set $d \leftarrow 1;$

Repeat

$d = (d + n)/e;$

Until d is integer

$e^{-1} \leftarrow d;$

Table 3.4 shows an implementation of the Baghdad algorithm.

Let $e \leftarrow 7; n \leftarrow 60;$	
d	*result*
(1+60)/7	not integer
(61+60)/7	not integer
(121+60)/7	not integer
(181+60)/7	not integer
(241+60)/7	integer match
$e^{-1} \leftarrow d = 43$	

Table 3.4: Result for Baghdad algorithm

3.2.5 **Fraction-Integer Method** (FIM)

This method outperforms the other methods for small number of e and irrelevant to n. As we can see that FIM algorithm is based only on addition which is the fastest operation, and that is why it outperforms the other methods except Euclidean for big numbers of e.

Algorithm

Input: e ∈ Z_n such that gcd (e, n) =1

Output: e^{-1} mod n, provided that it exists

1. Let d ←1.0 / e
2. err ← 1.0 / 2.0 * n)
3. Let def ← (double) n / e
4. cout << showpoint << fixed << setprecision(15)
5. do{
   - 5.1 d ← (d + def).
   - 5.2 cout << d << endl
6. while (d – (int) (d + err) > err)
7. cout << "The multiplicative inverse of ("<<e<<") % (" << (int) (d + err) << endl
8. e^{-1} ← d.

Table 3.5 shows an implementation of the Fraction-Integer algorithm.

Let e ← 7, n ← 60.

d	def
0.1429	8.5714
8.7143	
17.2857	
25.8571	
34.4286	
43.0000	

Table 3.5: Result for Fraction-Integer algorithm

$e^{-1} \leftarrow d = 43$

3.3 Fast Fraction-Integer Method (FFIM)

The idea behind the proposal method is very simple, based on continuous subtract fraction from integer and divide by fraction to obtain integer that will be used to compute the inverse d. The algorithm needs only six variables, and uses addition and division only. The complexity of the algorithm is $O(\log n)$

Algorithm

Input: $e \in Z_n$ such that $\gcd(e,n) = 1$

Output: $e^{-1} \bmod n$ provided that it exists.

The algorithm as follows:

$r : real;$

$i = 1;$

$s_f = (n + 1 \bmod e) / e;$

$d_f = (n \bmod e) / e;$

If $s_f = 0$ then

Stop;

Else

Repeat

$r = ((i - s_f) / d_f);$

$i = i + 1;$

Until r is integer

$d = (n * (r + 1)) + 1 / e;$

Table 3.1 shows an implementation of the FFIM algorithm.

Let $e \leftarrow 7; n \leftarrow 60;$

i	s_f	d_f	r
1	0.71428	0.57142	0.50001
2			2.25004
3			4.00000

Table 3.6: Result for FFIM algorithm

$d = (60 * (r + 1)) + 1 / e$

= (60*(4+1)) +1/7

= (60*5) +1/7

= 301/7

= 43

3.3.1 Proof of Fast Fraction-Integer Method

In order to prove the algorithm, we need to prove that the algorithm will give integer number only when d is the inverse of e. As we know that if d is the inverse of e then

1. Both e, d are positive integer numbers between $[1, n]$ ……. …… (1)

2. $\gcd(e, n) = 1$ ……..……...……... (2)

3. $e * d \equiv 1 \bmod n$, it means that $e * d = 1 + k * n$ for $k \in Z$ ……...…... (3)

So $d = (1 + k * n) / e$

$= 1/e + k * n / e$ …...…….... (4)

From the algorithm of Fast Fraction-Integer Method we see that $d = (n * (r + 1)) + 1/e;$ this will repeated i times until d ……..... (5)

From that we know that the algorithm above is correct for $i = k$, but if this is the case we need to prove that (5) will give non integer for all values of $i < k$, and the only integer value is when $i = k$, so we know d is an integer so $(1 + k * n) / e$ is also integer for integer value of k.

Then we need to prove that $(1 + i * n) / e$ is never an integer for all values of i between $[1, k - 1]$. Assume that there is another value of i where $i < i < k$ such that $d = (1 + i * n) / e$ is also an integer, it means that $i = k - 1$ ---------- (6)

Then $d = (1 + (k - 1) * n) / e$ will be integer. So

$$d = (1 + k * n - n) / e$$

$$= (1 + k * n) / e - n / e$$

$$= 1/e + k * n / e - n / e$$

We know that $1/e + k*n/e$ is integer, and also that $\gcd(e,n)$ should be 1. So if there is no greater common divisor between e and n except 1, that means n/e is a non integer value.

Thus subtracting a non integer value form an integer value will yield d is not an integer. This will contradict our assumption (that d is an integer).

Now assume that there exist $i = k - q$ such that d is an integer for q between $[1, k-1]$. Then

$$d = (1 + (k-q)*n)/e$$

$$= 1/e + k*n/e - q*n/e$$

If this to be integer then $q*n/e$ must be integer, but since $\gcd(e,n) = 1$ then q must be a multiple of e so $d = 1/e + k*n/e - x*n$ (5)

This will lead to d being a negative number $d < 0$ but from definition we know that both e and d must be positive (1) so there is no value for x that satisfy the definition. So the only value for q that satisfies the conditions is when $q = 0$ and that $i = k$.

3.3.2 Problem of Fast Fraction-Integer method

We have proved that Fast Fraction-Integer algorithm is correct, but the question is that is it implemental? Yes the algorithm will terminate giving the correct answer when implemented using the computer programming languages.

Let dm be the mathematical value of d where $d = dm$. Let dc be the calculated value of d in the computer memory and registers. Let ξ be the error in calculating, between the mathematical value and the computer value (round off error).

So $dm = (1m + km * nm)/em$

$$= 1m/em + km * nm/em$$

$$= (1/e)m + (k * n/e)m$$

But we know that the calculated value of fractions is never exactly as the mathematical value for big values of e that when used to divide 1 and n will give a cyclic fraction number.

So $(1/e)m = (1/e)c + \xi 1$ and $(n/e)m = (n/e)c + \xi 2$ where $\xi 1 << (1/e)c$, $\xi 2 << (n/e)c$ and $dc = (1/e)c + \xi 1 + k * \xi 2$ such errors will yield that either $dm \le dc$ or $dm \ge dc$, $dm - dc$ if and only if $\xi 1 + k * \xi 2 = 0$ it means that $(1/e)m = (1/e)c$, $(n/e)m = (n/e)c$ We know that the error $\xi 1, \xi 2$ is small, but multiplying $\xi 2$ with k will give big value to the error. The error will multiply by k, so as k is increasing the error. Also it will increase so the best approach is to use small values for e.

3.3.3. Compression between FFIM with Euclidean, and Baghdad methods.

Euclidean's algorithm is the oldest nontrivial algorithm and it still a good one because it approximately irrelevant to e or n, but the other algorithm is affected by e or n.

In (Sattar J., 2004) the author claims that Baghdad algorithm is more efficient and faster than existing methods except Euclidean's algorithm.

The proposed FFIM algorithm is more efficient than Baghdad algorithm.

Table 3.7 shows a number of iteration for implementation Euclidean, FFIM, FIM and Baghdad algorithms to find the multiplicative inverse d.

From table 3.7 we noticed that Baghdad algorithm and FIM algorithm take the same number of iterations to find d. We can summarize the amount of improvement in FFIM algorithm compared with Baghdad algorithm $= (e / (n \bmod e)) - (1 / (n \bmod e))$. (I.e. number of iterations in FFIM $* (e / (n \bmod e)) - (1 / n \bmod e)) =$ number of iterations in Baghdad algorithm.

n	e	d	Euclidean iterations	FFIM iterations	FIM iterations	Baghdad iterations
60	7	43	5	3	5	5
58200	967	13903	7	43	231	231
17460	17	16433	4	1	16	16
12610	571	4991	7	19	226	226
971	301	871	6	61	270	270
1870	301	1081	8	37	174	174
58200	53	48317	4	5	45	45
180	31	151	4	21	26	26
3233	402	2662	6	14	331	331
58200	7	24943	3	1	3	3

Table 3.7: Number of iteration in Euclidean, FFIM, FIM and Baghdad methods

Figure 3.1 show the comparison between the FFIM algorithm with Euclidean algorithm and Baghdad algorithm. We have implemented the algorithm for value in table 3.7; where x axis represents the values of (e, n) and y axis represent number of iterations.

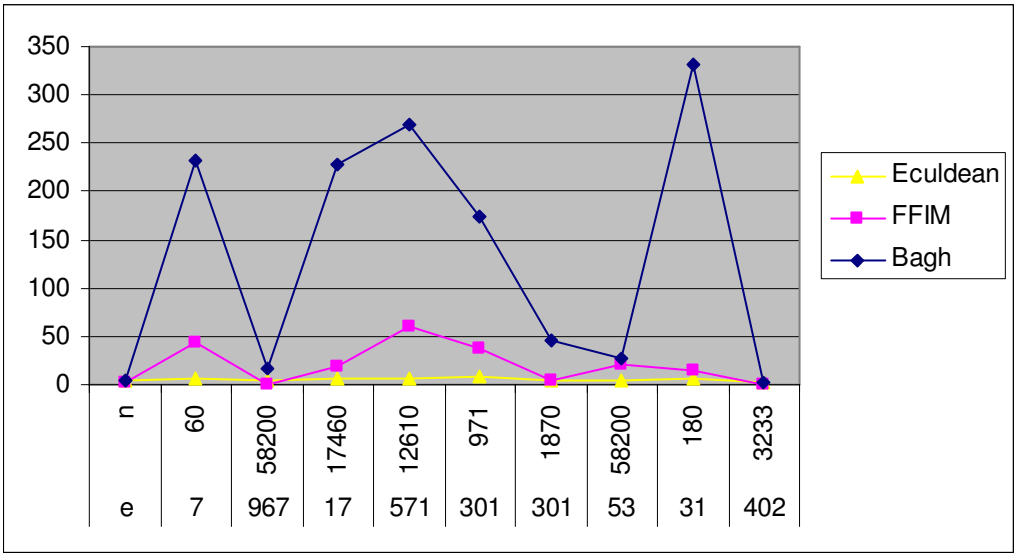


Figure 3.1: comparison between the FFIM algorithm with Euclidean and Baghdad algorithms.

Chapter 4

Anonymous and Non-Repudiation E-Cash Scheme based on Partially Blind Signature

4.1 Introduction

The general e-cash scheme which is appropriate for both the existing protocols and the suggested protocol includes five entities: a Customer, a Blind Office, a Bank , a judge, and a Payee.

- ***Customer*:** purchases goods or services from merchant using e-cash
- ***Bank*:** issues e-cash and maintains bank account for customers and merchants.
- ***Blind Office*:** pseudo identity escrow agency, it can link all payments made under the same PID
- ***Payee*:** sells goods or services to customer, and deposits e-cash to bank
- ***Judge*:** to adjudicate between the three entities; Bank, Customer and Blind Office.

These five entities act as follows:

When Customer decides to withdraw coin he/she obtains coin blindly signed by Bank. Bank holds a relation proves between customer's real identifier (ID) and pseudo identifier (PID), Blind Office participated in the blind signature, preserves another relation prove among PID and Coin. To spend Coin, Customer proofs to Payee that he has ken of secret key x accordance to Coin. If Coin is misused, for example over spend, Bank and Blind Office will cooperate to make a link amongst ID and Coin, J will participate in these tracing steps to judge.

Figure 4.1 shows the coin's three fields:

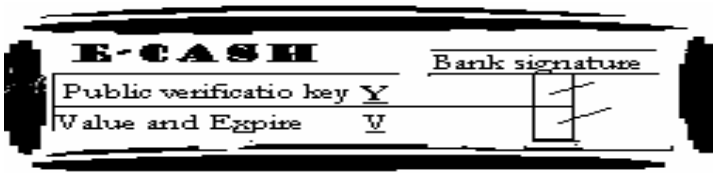


Figure 4.1 coin fields.

- The public verification key denoted by y for a public key type signature scheme. The corresponding secret signature key is represented by x.
- The data item v (pre-define message between Bank and Customer) having some pertinent data concerning Coin, for example its value and expiry dates.
- The bank's digital signature on both y and v.

In M'raihi's scheme (1995) based on blind signatures, blinding is sub contracted to a trustee, using identity-linked pseudonyms (PIDs) to achieve anonymity. The bank acts as a Certification Authority that provides the Customer with the certified PIDs (therefore the bank can link the pseudonym to the real identity of the Customer). The trustee is designed as a Blinding Office, it can link all payments made under the same PID. The Customer may use the PIDs to pseudo-identify himself to the Blind Office.

This scheme permits both bank and Blind Office to playact a customer without being noticed (see 4.2.3).

Aboud & Fayoumi (2007) based on blind signature, suggested different scheme to prevent the blind office and bank from playact as a customer, so that the customer cannot disclaim it when misused coin. These benefits are at the Customer computational cost and are more burdensome than for the existing protocols, since the Customer requires to make pre- calculations of the digital signature scheme when employing digital signature scheme for spending Coin . In this scheme employ the Diffie-Hellman scheme to generate shared w.

In this chapter we will introduce a non repudiation e-cash scheme based on partially blind signature which enables Judge to specify a dishonest Customer, bank, or Blind Office. Also the bank needs only to keep the still-alive e-cashes in the database to prevent double spending in on-line scenario. Our scheme provides the three characteristics anonymity, non repudiation and traceability in off-line scenario.

This chapter is organized as follows. In section 4.2 we explain the M'Raihi Scheme. In section 4.3 we described the propose scheme. Section 4.4 illustrates the scheme correctness. The section 4.5 illustrates the security of the propose scheme. Finally we state concluding remarks in section 5.

4.2 The M'Raihi Scheme

In this scheme the bank acts as a Certification Authority (CA) that provides the customer the certified PIDs (therefore the bank can link the pseudonym to the real identity of the customer). The trustee is designed as a Blinding Office (BO), it can link all payments made under the same PID. The customer may use the PIDs to pseudo-identify himself to the BO.

This e-cash scheme consists of five entities, Customer, Bank, Blind Office, Payee and judge.

The M'Raihi scheme acts as follows. Customer and Bank first set up a shared private s. Bank then signs a free one-way hash function of s, that is $S_{Bank}(h(s))$, which is employed to build PID by concatenating it with $E_{BO}(s)$, Bank also holds relations among ID and PID, which we named by {ID, PID}.

4.2.1 Withdrawal Protocol

Customer demonstrates Blind Office both PID and x which is created by Customer. Blind Office calculates a related y and a set of pre-calculated values. BO then blinds y with a random blinding factor w to find $\bar{y} = f(w)y$. Bank signs $\bar{y}$ without knowing y and withdraws a true coin from Customer's account.

Blind Office obtains Coin from Bank's signature on $\bar{y}$ and offers it to Customer. Blind Office preserves a relation among PID and Coin, which we denote by {PID, Coin}.

4.2.2 Spend protocol

To spend Coin, a Customer sign a message, which is created by Payee as a challenge, to prove Customer knows x. Payee demands a true coin back from Bank. If Coin is over spending, Bank will inquire for tracing steps in which Bank and Blind Office work together to construct a relation among Coin and ID, depending on {ID, PID} and {PID, Coin}.

4.2.3 Vulnerabilities and cryptanalysis

M'Raihi scheme (1995) is based on high trust relations between a Customer, Bank and BO. Also Bank and BO should be trusted not to playact a Customer to get and spend Coin, since they are able to do thus when they desire. Throughout tracing method, Customer can perform more than one demand to Judge to suggest that Bank or BO have been playact a Customer.

- Blind Office is able to playact a Customer to spend Coin. It happens since Blind Office knows x.
- Bank is able to playact a Customer to Blind Office to get Coin, and is then capable of playacting a Customer to Payee to spend Coin. This happens since Bank knows s.

4.3 Proposed Scheme

To build e-cash scheme, such scheme must satisfy the following properties (Chaum 1998), (Fan, et al., 2000),(Shao, 2000)

***Correctness*:** the correctness of the signature of a message signed through the signature scheme can be checked by anyone using the signer's public key.

***Authenticity*:** a valid signature implies that the signer deliberately signed the associated message.

***Unforgeability*:** only the signer can give a valid signature for the associated message.

Non-re-usability: the signature of a document can not be used on another document.

Non-repudiation: the signer can not deny having signed a document that has valid signature.

***Integrity*:** ensure the contents have not been modified.

***Blindness*:** the content of the message should be blind to the signer; the signer of the blind signature does not see the content of the message.

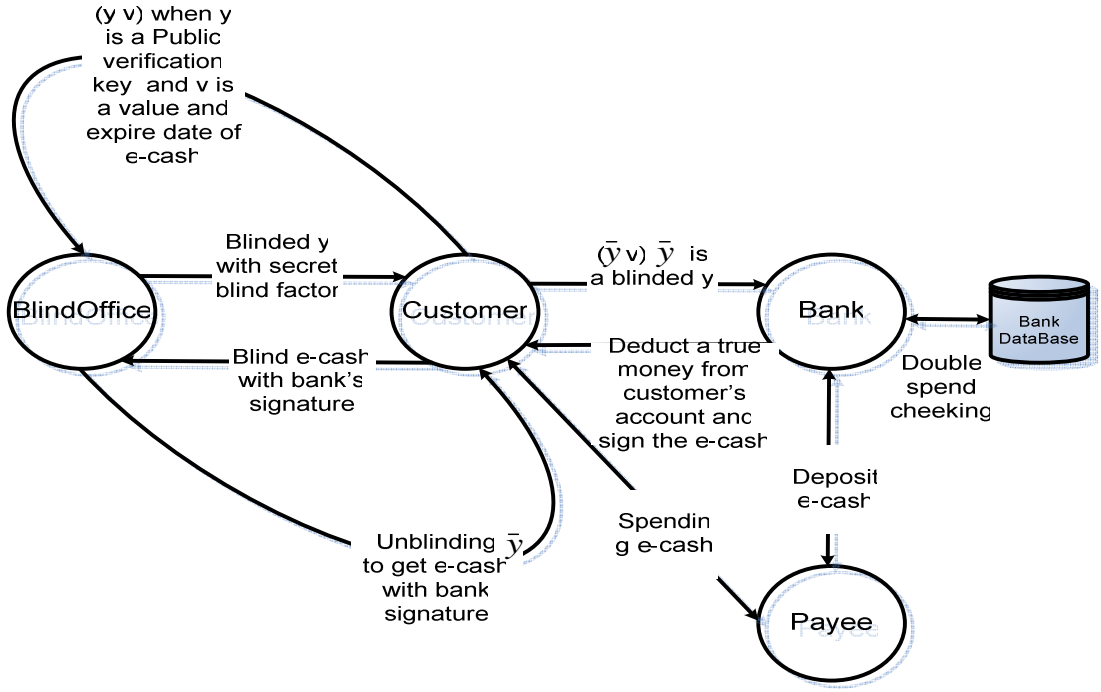
***Untraceability*:** the signer of the blind signature is unable to link the message-signature pair even when the signature has been revealed to the public.

Our electronic cash scheme consists of five entities, *Customer, Bank, Blind Office, Payee and Judge*. Our scheme works as follows:

Customer obtains a coin signed *partially blind* by Bank to allow the Bank to explicitly include some agreed information in the blind signature. Using partially blind signatures in e-cash system, we can prevent the bank's database from growing unlimitedly. Then Bank holds a relation proof among customer real identifier ID and pseudo identifier PID. Blind Office BO participated in a partially blind signature, holds another relation proof among customer pseudo identifier PID and coins. To spend coins, customer need to prove to payee that he/she know x. If coin is misused, for example double spending, we have two scenarios; in on-line scenario we well go to double spending prevention case to ensure that no coin is spent twice, in off-line

scenario we well go to trace step which requires bank and BO to work together to construct a link among ID and coin, and judge will participate in this procedure to judge.

This procedure is depicted in figure 4.2.



(Figure 4.2) proposed e-cash scheme

Customer and Bank both have exponent key type signatures denoted by (S_{cust}, V_{cust}) and (S_{Bank}, V_{Bank}) respectively, such that V_{cust} is known to *Bank and* V_{Bank} *is* known to *customer*, *BO* and *Payee*.

Assume also that *BO* has an exponent public-key cryptosystem, denoted by (E_{BO}, D_{BO}) such that E_{BO} *is* known to *customer* and Bank, *Judge* could be verified all the schemes.

The potential implementation for these cryptosystems is RSA Scheme (Rives, et al., 1978). In fact, we need the signature scheme of the bank to have some particular property, that is S_{Bank} (m1) S_{Bank} (m2) = S_{Bank} (m1 m2), which holds for RSA. Certainly this is usually a largely unfavorable feature for a signature scheme, and is one reason why RSA must always be employed in combination with a special redundancy function or a one-way hash function (Menezes, 1997) (Stinson, 2006). We indicate a blinding function by BF, and allow it be an

inverse of the signature scheme, so that S_{Bank} (BF (m1) m2) = m1 S_{Bank} (m2) for each m1, m2.

Never use RSA to sign a random document presented to you by a stranger. Always use a one-way hash function first (Schneier, 1996).

We will refer to a blinding function by *BF*, and use it as an inverse to the digital signature scheme.

If Bank's signature scheme is RSA, then *BF* is just exponentiation employing the public verification key.

The value of v created by *Customer* is a message predefined by the *Bank* which is considered as clear part. Previous scenario will be achieved via three protocols as follows:

4.3.1 The Initial Protocol

The bank publishes: (*n, e, f*), where n and e is a standard RSA public key and the private key *d* is the multiplicative inverse of *e* can be calculated by FFIM (see section 3.3) .

And f is an appropriate public exponent generating function. With RSA based partially blind signature scenario $f()$ a public exponent function satisfying

- $f(v)$ must be different for different values of *v*; where *v* is a message predefined by the bank and contains an expiration date of the e-cash and BO's name.
- $f(v)$ must be relatively prime to (n).
- Each output has at least one unique prime factor.

The Customer and Bank generate a shared secret s.

Then Bank signs a one way hash function of s, namely S_{Bank} (h(s)) which is employed to build PID by concatenating it with E_{BO} (s).

4.3.2 The Withdraw Protocol

The withdraw protocol can be divided into three phases.

The first phase is called blinding phase as shown in figure 4.3, when a customer decides to withdraw e-cash from the bank, he/she randomly chooses x and calculate y then sends $E_{BO}(y, v)$ to BO where v is a message predefined of by the bank and contains an expiration date of the e-cash, value, and BO's name.

The BO chooses blind factor r and calculates $\bar{y}=BF(r, f(v)).y$ and passes $\bar{y}$ to customer.

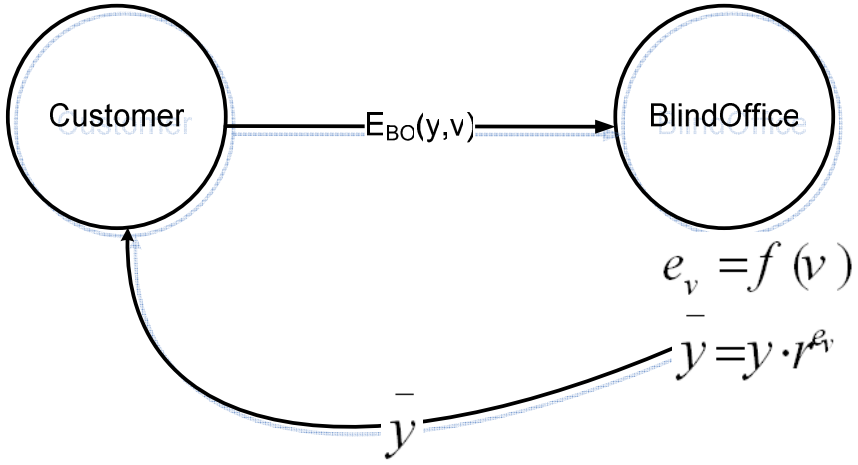


Figure 4.3 Withdrawal protocol (blinding phase)

The second phase is called signing phase as shown in figure 4.4. The customer then sends $\bar{y}$ and v to bank. The bank verifies whether or not v is correct If it is correct the bank calculates $t= E_{BO}(S_{Bank}(\bar{y}))$ and sends t to customer and deducts true coins from customer's account and keeps a relation proof {ID, $\bar{y}$}.

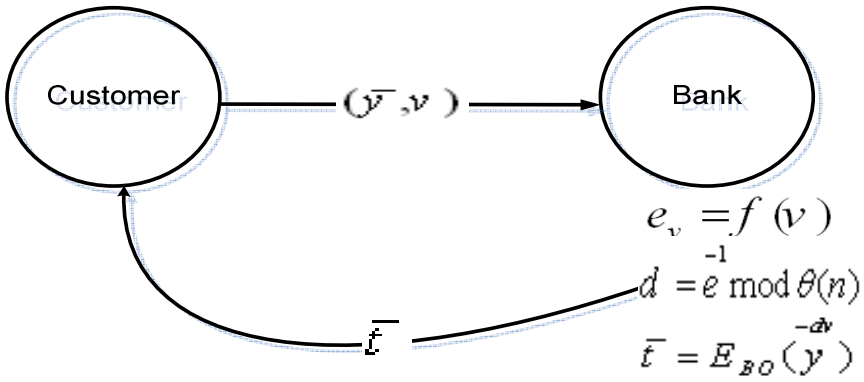


Figure 4.4 Withdrawal protocol (signing phase)

The unblinding phase is the third phase in withdrawal protocol, the customer passes $\bar{t}$ to BO. Then BO recovers $\bar{t}$ to get $S_B(\bar{y})$ then unblinds $S_{Bank}(y)$ then builds coins and passes it to customer and holds a relation {PID,Coins} as shown in figure 4.5.

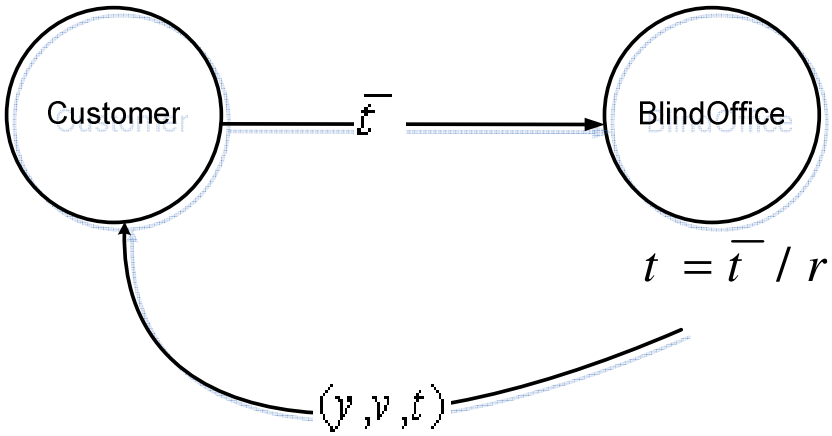


Figure 4.5 Withdrawal protocol (unblinding phase)

4.3.4 The Spending Protocol

The spending protocol as shown in Figure 4.6. The payee creates a challenge message to prove customer knows x.

Payee claims a true coin back from Bank later.

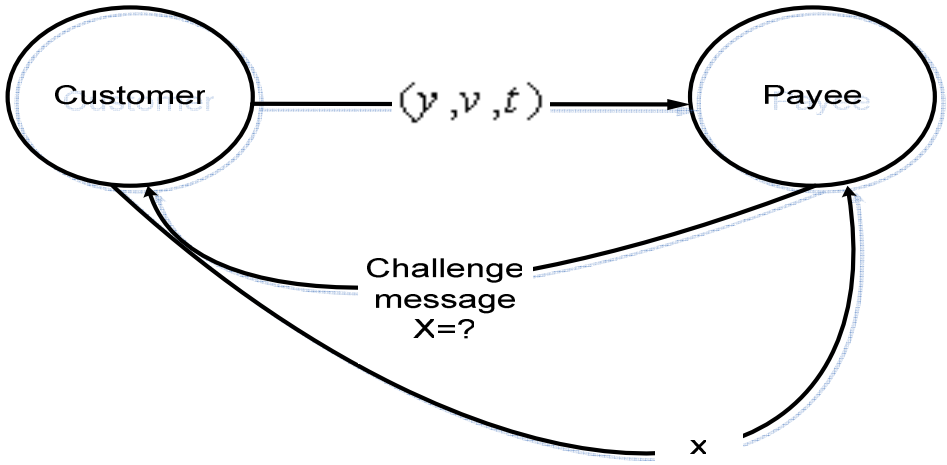


Figure 4.6 Deposit protocol

4.3.5 Double Spending occur

If C is double spending, Bank will request tracing steps in which Bank and BO work together to construct a link among Coin and ID , relying on {ID, PID} and {PID,Coin}.

4.4 Correctness

- **Partially blinding phase**

  $e_v = f(v)$

  $\bar{y} = y \cdot r^{e_v} \bmod n$.....(1)

- **Signing Phase:**

  $d_v = \bar{e}_v^{-1} \bmod \theta(n)$.

  $\bar{t} = \bar{y}^{d_v} = (y \cdot r^{e_v})^{d_v} = y^{d_v} \cdot r \bmod(n)$.....(2)

- **Unblinding Phase**

  $t = \bar{t} \square r^{-1}$.....(3)

- **Verification Phase**

  Theorem 1. If (*v, y,* t) is a signature of the message *y* produced by the proposed scheme, then $y \equiv t^{e_v} \bmod n$

  Proof.

  $$t^{e_v} \stackrel{?}{\equiv} y \bmod n$$

  $$t^{e_v} \equiv (r^{-1} \cdot \bar{t})^{e_v} \text{..from 3}$$

  $$\equiv (r^{-1} \cdot \bar{y}^{d_v})^{e_v} \text{..from 2}$$

  $$\equiv (r^{-1} \cdot (r^{e_v} \cdot y)^{d_v})^{e_v} \text{..from 1}$$

  $$\equiv (r^{-1} \cdot r \cdot y^{d_v})^{e_v}$$

  $$\equiv (y^{d_v})^{e_v}$$

  $$\equiv y \bmod n$$

4.5 Security of the scheme

In our scheme we claim that customer, BO and bank can't repudiate any steps done by him. Also the entities in our scheme can not impersonate each other: The other important benefit that the bank needs is only to keep the still-alive e-cashes in the database to prevent double spending in on-line scenario.

We now plan the proofs that the our scheme owns this security abilities

Theorem 1: customer can not get Coin without the participation of Bank and BO.

Proof 1: to get Coin without Bank or BO be included, customer should be capable of calculating $S_{Bank}(y)$ or $E_{BO}(S_B(y))$, each of which is supposed to be infeasible.

Theorem 2: Neither Customer nor BO can change value of v after U sends it to BO.

Proof 2: To change value of v to v1 BO needs to calculate S_U on v1 because the bank's signature on $\bar{y}$ depends on v that was sent by customer .Then unblinded processes can not be achieved. If customer changes value of v to v1 and sends it to bank, this change must be made in $\bar{y}$ that bank will sign on it. And in unblind processes the BO will be unable to get $S_{Bank}(y)$ with new value.

Theorem 3: The member who is the publisher of x can just spend a valid coin relevant to a secret key x.

Proof 3: Suppose that given y and further associated public data, it is computationally infeasible to decrypt x, x is given only to its publisher and is not exposed to any other person. So, because knowing of x is needed to spend Coin, the finding follows.

Theorem 4: The BO cannot impersonate a customer to Bank.

Proof 4: To impersonate Customer to Bank, Blind Office should find *Customer's* digital signature

on the $\bar{y}$ and v, which is supposed to be infeasible.

Theorem 5: When Bank impersonates a customer to get and to spend a Coin, he cannot say that Customer published the coin.

Proof 5: To verify {ID, $\bar{y}$}, Bank wants Customer's signature on $\bar{y}$ and v. Such a signature cannot be found, in spite of BO's cooperation.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

For security reasons, cryptography recommends smaller values for public keys and bigger values for private keys (Menezes, Oorschot, and Vanstone, 1996) The FFIM proposed algorithm for computing multiplicative inverse needs lower values for public keys (lower value of e) and higher values for private key, which is fully compatible with the preferred cryptography algorithm. The method is simple, fast and needs less storage, and its complexity is also less.

Our proposed e-cash scheme with partially blind signature designed to provide characteristics of anonymity, non-repudiation and traceability.

The proposed scheme is suitable for on-line; the bank assures that the signed e-cash carry the agreed common information- expiration date. With this common information, the bank needs only to keep the still-alive e-cashes in the database to prevent double spending.

The proposed scheme is multi-purpose, as it allows the integration of multi spendable and divisible coins.

5.2 Future Work

In many e-commerce environments, the client sides will be the mobile stations or the smart cards. Due to the computation constraint of these devices, low-computation protocol design is urgently needed. In view of this requirement, Fan and Lei (1998) proposed their low-computation partially blind signature, based on Rabin scheme (Rabin, 1979). Their scheme reduces the amount of computation on the client side by almost 98%.

As a future work, we would like to focus more on developing the proposed scheme to make it attractive for mobile client and smart-card implementation.

References

1. Abe, M. and Fujisaki, E. (1996). **How to Date Blind Signatures**. Advances in Cryptology: ASIACRYPT '96: 244-251.

2. Abe M. and Camenisch. J. (1997). **Partially blind signatures**. Symposium on Cryptography and Information Security.

3. Aboud, S. J. (2004). **Baghdad Method for calculating Multiplicative Inverse**, The International Conference on Information Technology (ITCC 2004), IEEE, Las Vegas, U .S. A.

4. Aboud, S. J. Abu-Ayyash, A. (2005). **Fraction-Integer Method**, the 7th World Multiconference on Systemics, Cybernetics and InformaticsOrlando,Florida, USA, July 27 – 30.

5. Aboud, S. J. and AL-Fayoumi M. A. (2007). **Anonymous and Non-Repudiation E-Payment Protocol**. American Journal of Applied Sciences 4 (8): 538-542.

6. Asokan, N. Janson, P. Steiner, M. and Waidner. M. (1997) **State of the art in electronic payment systems**. IEEE Computer, 30(9):28-35.

7. Boly, J. Bosselaers, A. Cramer, R. Michelsen, R. Mjolsnes, S. Muller, F. Pedersen, T. Pfitzmann, B. de Rooij, P. Schoenmakers, B. Schunten, M. Vallee, L. Waidner, M. (1994). **The ESPRIT Project CAFE - High Security Digital Payment Systems**, *ESORICS '94 Proceedings*, pp. 217-30, Lecture Notes in Computer Science vol. 875. Springer-Verlag, Berlin.

8. Chaum, D. (1982). **Blind signatures for untraceable payments**. In D. Chaum, R. Rivest, and A. Sherman, editors, Advances in Cryptology | Proceedings of Crypto '82, pages 199{204. Prenum Publishing Corporation.

9. Chaum, D. (1983) **Blind Signature for Untraceable Payments**, Advances in Cryptology. Crypto'82. Plenum Press.

10. Chaum, D. (1985) **Security without identification**: Transaction systems to make big brother obsolete. Communications of the ACM, 28(10):1030–1044.

11. Chaum, D(1989). **Online cash checks**. In Jean-Jacques Quisquater and Joos Vandewalle, editors, Advances in Cryptology – EUROCRYPT '89, volume 434 of LNCS, pages 289–293.

12. Chaum, D. Fiat, A., and Naor, M. (1988) **Untraceable electronic cash**. In Shafi Goldwasser, editor, Advances in Cryptology – CRYPTO '90, volume 403 of LNCS, pages 319–327.

13. Chaum, D. (1998). **Blind signatures for untraceable payments**. Advances in cryptology, CRYPTO'82, Lect. Notes Computer Science, (Springer-Verlag, 1998), pp. 199-203

14. Camenish, J. Maurer U. and Stadler, M.(1996). **Digital Payment Systems with Passive Anonymity- Revoking Trustees**, Computer Security - ESORICS 96.

15. Claessens, J. Preneel, B. and Vandewalle, J. (1999). **Anonymity controlled electronic payment systems**, *20th Symposium on Information Theory in the Benelux*, Haasrode, Belgium, pp. 109-116.

16. Davies, G. (1996) **A history of money from ancient times to the present day.** University of Wales Press, Cardiff.

17. Davida, G.I. Frankel, Y. Tsiounis, Y. and Yung, M. (1997). **Anonymity Control in E-Cash Systems**, Financial Cryptography 97.

18. ElGamal, T. (1985). **A public key cryptosystem and signature scheme based on discrete logarithms**, IEEE Transactions on Information Theory.

19. Fan, C.I. and Lei, C.L. (1998). **Low-computation partially blind signatures for electronic cash**, IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences, vol. 81, no. 5, pp. 818-824.

20. Fan, C.-I. Chen, W.K. and Yeh, Y. S. (2000). **Randomization enhanced Chaum's blind signature scheme**, Computer Communications, vol. 23, pp. 1677–1680.

21. Gordon, J. (1989). *Fast* **Multiplicative inverse in modular arithmetic**, cryptography and coding, clarendon press oxford.

22. Hwang, M.S., Lee, C.C. and Lai, Y.C. (2002). **Traceability on low computation partially blind signatures for electronic cash**. IEICE Trans. Fundam, 85: 1181-1182. http://search.ieice.org/bin/summary.php?id=e85-a_5_1181&category=A &lang=&year=2002&auth=1.

23. Huang,H.-F. and Chang, C.-C., (2004). **A new design of efficient partially blind signature scheme**. J. Syst. Software, 73: 397-403. doi:10.1016/S0164-1212(03)00237-1.

24. Juang, W.-S. Lei, C.-L.and Chang, C.-Y. (1999). **Anonymous channel and authentication in wireless communications**, Computer Communications *22*, pp. 1502-1511.

25. Jakobsson M. and M'Raïhi, D.(1998). **Mix-based Electronic Payments**, Fifth Annual Workshop on Selected Areas in Cryptography (SAC '98), Queen's University, Kingston, Ontario, Canada, August 17-18.

26. Kahn 1967 D. Kahn, *The code breakers: the story of secret writing*. Scribner (New. York, Macmillan).

27. Knuth, D. E. (1981), "**The art of computer programming**, Vol. 2 semi numerical algorithms', 2nd Ed., Addison-Wesley.

28. Liu, J., K. Wei and S. Wong, (2001). **Recoverable and Untraceable E-Cash**, EUROCON'2001: Trends in Communications. Intl. Conference on Information Technology, Vol. (1): 342-349.

29. Rabin, M. O. (1979). **Digitalized signatures and public-key functions as intractable as factorization, Technical Report**, MIT/LCS/TR212, MIT Lab., Computer Science, Cambridge,Mass.

30. Menezes, P.C. Van Oorschot, and Vanstone, S.A. (1996). **Handbook of applied cryptography,** p67, p71.

31. Miyazaki, S., Abe, M. and Sakurai K. (1997). **Partially blind signature schemes for the DSS and for a discrete log. based message recovery signature**. In Proceedings of the 1997 Korea-Japan Joint Workshop on Information Security and Cryptology

32. Medvinsky, G. Neuman, B. C. (1993). **NetCash: A Design for Practical Electronic Currency on the Internet**, In ACM-CCS'93.

33. M'Raihi D, (1996). **Cost Effective Payment Schemes with Piracy Regulation**, Advances in Cryptology-ASIACRYPT 96, Lecture Notes in Computer Science 1163. Springer-Verlag, pp: 266-275.

34. M'Raïhi, D. and Pointcheval, D. (1998). **Distributed Trustees and Revocability: a Framework for Internet Payment**, In Financial Cryptography '98, Springer-Verlag, LNCS 1465.

35. O'Mahony, D. Peirce, M. and Tewari, H. (1997). **Electronic payment systems**. Artech House, Boston/London.

36. Okamoto, T. (1994). **designated confirmer and public encryption are equivalent**, advance in crypto-crypto-94, springer and Verlag, LNCS # 893,p.p.61-74.

37. Peirce, M. (2000). **Multi-Party Electronic Payments for Mobile Communications. PhD thesis**.

38. Rivest, R., A. Shmir and L. Adlman, (1978). **A Method for Obtaining Digital Signatures and Public Key Cryptosystems**. Communications of the ACM (21): 294-299.

39. Schneier, B (1996). **Applied Cryptography**, 2ed ed., John Wiley and sons. p246.

40. Solms, V and D. Naccache, (1992). **Blind Signatures and Perfect Crimes**. Intl. J. Computers & Security, Vol. (11): 581-583.

41. Susan Hohenberger, (2006). **Advances in Signatures Encryption and E-Cash from Bilinear Group**. PhD Thesis. MIT, pp. 95-142.

42. Stein, J. (1967). Stein, J. (1967) J. Comp. Phys, 1, p397-405.

43. Shao, Z. (2000). **Improved user efficient blind signatures**, Electronics Letters, vol. 36, no. 16, pp. 1372–1374.

44. Zhang, F. and Chen, X. (2005). **Cryptanalysis of Huang-Chang partially blind signature scheme**. J. Syst. Software, 76: 323-325. doi: 10.1016/j.jss.07.249.

APPENDICES

APPENDIX A: CURRICULUM VITAE

Hani M. AL-Matari

Personal Information

Date of Birth : July 3, 1983

Place of Birth : Sana'a

Nationality : Yemen

Marital Status : married

Mobile :

E- mail : Hani_almatari@hotmail.com

Objectives

I am seeking a challenging post as a lecturer, in an academic staff university that has the need to an active academic staff, for information technology college

Summary of Qualifications

Programming Languages:

- C#:- I have many courses of C# and working with it proficiently and having many projects on it such as my graduation project
- Vb.net:- I have an experience for more than 6 month programming using vb.net and have Many projects
- ASP.net
- Java (J2SE)
- prolog,
- C++ and Object oriented programming

Databases:

- Microsoft server DB: - working with it proficiently SQL server as ADO.net and all management tools
- MySQL DB: - can build database by MySQL
- MS Access

Other Skills

- Microsoft developing web application
- maintenance of hardware and software

- Microsoft Office 2000,2003, 2007
- Video maker
- Macromedia Flash MX
- HTML:- using HTML to Deign a webpage

CERTIFICATIONS
Cisco Certified Network Associate (**CCNA**)
Oracle Certified Professional (**OCP**)

Education

M. S. Computer Science, Middle East University for Graduate Studies, Faculty of Information Technology, **Jordan-Amman** Jan, 2011.
Overall Grade: **Excellent.**

M. A. Thesis
Anonymous and Non-Repudiation E-Cash Scheme based on Partially Blind Signature

B. S. Computer Science Applied Science University, Faculty of Information Technology, **Jordan-Amman**. Jan, 2007
Overall Grade: **Very Good.**

Experiences

B. S. Graduation project

Internet Service Provider Management System

A system to work together with an ISP RADIUS system, to manage customers' information, create reports easily, and to manage employee information. To be used by the ISP's database administrators, management members, and technical support team.

* This project is to be used in Applied Science University to manage the Internet subscribers.

Publication

2009 Hani M. AL-Matari, Sattar J. Aboud and Nidal F. Shilbayeh "**Fast Fraction-Integer Method for Computing Multiplicative Inverse**" JOURNAL OF COMPUTING, VOLUME 1, ISSUE 1, DECEMBER 2009, ISSN: 2151-9617
http://www.journalofcomputing.org/volume-1-issue-1-ember-2009

2011 Hani M. AL-Matari and Nidal F. Shilbayeh. **"Anonymous and Non-Repudiation E-Cash Scheme based on Partially Blind Signature**" JOURNAL OF COMPUTING, VOLUME 3, ISSUE 2, February 2011, ISSN: 2151-9617
http://www.journalofcomputing.org/volume-3-issue-2-february-2011

Languages

Arabic: mother language.

English: excellent command in both spoken and written

I. PERSONAL QUALITIES

- Reliable and able to work under pressure and in difficult work circumstances.
- Punctual and very adequate in deadlines.
- Good teamwork spirit.
- Excellent communication skills with work colleagues as well as with other people.
- Have the ability to understand the Request of success, and Leadership arts.

Interests

Working, Reading, Solve Puzzles.

APPENDIX B: MATHMATICAL BACKGROUND

B.1 Introduction

Cryptography has always relied on mathematics of some sort; the main purpose of cryptography is to protect the interests of parties communicating in the presence of adversaries. A cryptosystem is a mechanism or scheme employed for the purpose of providing such protection. We examine several cryptosystems in this paper, spanning a wide range of cryptographic uses.

B.2 Digital signing

The digital signature provides a means to electronically replace a handwritten signature. Each signature must be associated, with high probability, with one particular person and one particular document. A digital signature can also act as proof of identity because only the person possessing the correct private key can generate signatures verified by the corresponding public key. Digital signature is a mechanism by which a message is authenticated i.e. proving that a message is effectively coming from a given sender, much like a signature on a paper document.

Digital signing is closely related to public-key cryptography. For messages sent through an insecure channel, a properly implemented digital signature gives the receiver reason to believe the message was sent by the claimed sender. Many publickey cryptosystems can also be used as digital signature system by simply reversing the order of operations: \encrypt" using the private key to generate the signature, and verify it by \decrypting" with the public key; this works because the operations are inverses of each other. Only the possessor of the particular private key can generate a signature that is correctly verified by the corresponding public key, and the signature for each document is different (again, with high probability).

B.3 Pseudorandom number generation

It is an algorithm for generating a sequence of numbers that must appear random. The source of randomness is of crucial importance for many cryptographic applications. Because natural

randomness is somewhat difficult to come by in large amounts, it is important to design pseudorandom number generators to supply numbers that appear to be random. The appearance of randomness is usually defined by the difficulty of predicting the next number (or bit), given the ones produced so far.

B.4 Complexity theory

Complexity theory concerns itself with the computational complexity of algorithms, giving us an upper bound on their computing requirement. The analysis of computational resources required to solve problems is the realm of complexity theory, pioneered in 1965 by Hartmanis and Stearns(1965). When determining the complexity of an algorithm, the salient measure is the asymptotic time or space required of an algorithm in terms of some size parameter n of the input. An algorithm runs in, say, $O(n^2)$ time (pronounced \big-oh of n squared") if its running time can be bounded asymptotically by some constant multiple of n^2. In general, the set of functions f(n) obeying a particular asymptotic bound g(n) can be denoted as follows:

O (g(n)) = { f(n) : there exist positive constants c and n_0 such that $0 \leq f(n) \leq cg(n)$ for all $n \geq n_0$g} Leiserson, Rivest, and Stein (2001).

B.5 One-way hash functions

Informally, a one-way function is a function that is easy to compute but difficult to invert. We are concerned primarily with cryptographically relevant one-way functions, and these tend to fall into two major categories: one-way hash functions and trapdoor functions. We will discuss trapdoor functions in the context of public-key cryptography, but here we introduce one-way hash functions.

Various other properties are sometimes associated with the concept of a one-way function, such as that it must be one-to-one or that it must be honest, meaning that for any x in the domain, f(x)

may be no more than polynomial smaller than x. A one-way hash function used in cryptographic applications, such as MD5 or SHA, generally has neither of these properties. Its purpose is to create a smaller, usually fixed-size value so that it is difficult to find a message that hashes to any particular value, or even any two messages that hash to the same value. Because the message space tends to be much bigger than the space of hash values, the hash function is not one-to-one, and it clearly cannot be honest with a fixed-size output. For a detailed look at cryptographic one-way hash functions, including myriad real-world examples (Schneier's, 1996). Although there are differing opinions on just what should constitute a one-way function, we will attempt to make some generalizations and draw conclusions relevant to cryptography.

```
#include <iostream>
#include <fstream>
#include <math.h>

using namespace std;
//int gcd (int ,int );

ofstream FFIM ("c:\\FFIM.txt");

double Round(double Value, int NumPlaces)
   {
   int k, Temp;
   float Factor;

   Factor = 1;
   for (k = 0; k < NumPlaces; k++)
      Factor = Factor * 10;

   Temp = Value * Factor + 0.5;
   return Temp / Factor;
   }


void bagh (int w , int c)
{

      double sub_fraction , div_fraction, r, result,sub1,div1;
      int r1,i;
      result=1;
      i=1;

      sub1 = (((c+1) % w)) ;
      div1 = ((c %w));

       sub_fraction = sub1/w;
       div_fraction = div1/w;

      cout<<sub_fraction<<endl;
      cout<<div_fraction<<endl;
      cout<<sub1<<endl;
      cout<<div1<<endl;

      if (FFIM.is_open())
      {


      FFIM<<"e"<<'\t'<<"n"<<'\t'<<"r"<<'\t'<<"result"<<'\t'<<'\t'<<"i"<<endl;

      while (result != 0)
      {
            r= (i- sub_fraction)/div_fraction;
            r1=r;
            result = Round(r,5) - r1;
            cout<<w<<'\t'<<c<<'\t'<<r<<'\t'<<result<<'\t'<<'\t'<<i<<endl;

            FFIM<<w<<'\t'<<c<<'\t'<<r<<'\t'<<result<<'\t'<<'\t'<<i<<endl;
            i=i+1;

            if(result == 1)
                  break;
```

```
      }

      cout<<"d =" << ((c * (r + 1)) +1) / w <<endl;
      FFIM<<"d =" << ((c * (r + 1)) +1) / w <<endl;

      FFIM.close ();


      }
}



int gcd (int n,int e)

{
      int r;

      while ( e >0 )
  {
      r = n % e ;
      n = e ;
      e = r ;
  }


  return n;
}

void main ()
{
      int w, n;
      cout<<"pleas enter e: ";
      cin>>w;
      cout<<endl;
      cout<<"pleas enter Q: ";
    cin>>n;



      if (gcd(w,n)==1)
      {
            bagh(w,n);
      }
      else
            cout<<gcd(w,n)<<endl;

}
      cout<<"d =" << ((c * (r + 1)) +1) / w <<endl;
      FFIM<<"d =" << ((c * (r + 1)) +1) / w <<endl;

      FFIM.close ();


      }
}
```

```
int gcd (int n,int e)

{
      int r;

      while ( e >0 )
  {
      r = n % e ;
      n = e ;
      e = r ;
  }


  return n;
}

void main ()
{
      int w, n;
      cout<<"pleas enter e: ";
      cin>>w;
      cout<<endl;
      cout<<"pleas enter Q: ";
    cin>>n;



      if (gcd(w,n)==1)
      {
            bagh(w,n);
      }
      else
            cout<<gcd(w,n)<<endl;
}
```

APPENDIX C: Publication

JOURNAL OF COMPUTING, VOLUME 1, ISSUE 1, DECEMBER 2009, ISSN: 2151-9617
HTTPS://SITES.GOOGLE.COM/SITE/JOURNALOFCOMPUTING/

Fast Fraction-Integer Method for Computing Multiplicative Inverse

Hani M. AL-Matari[1] and Sattar J. Aboud[2] and Nidal F. Shilbayeh[1]
[1] Middle East University for Graduate Studies, Faculty of IT, Jordan-Amman
[2] Information Technology Advisor, Iraqi Council of Representatives, Baghdad-Iraq

Abstract - Multiplicative inverse is a crucial operation in public key cryptography, and been widely used in cryptography. Public key cryptography has given rise to such a need, in which we need to generate a related public and private pair of numbers, each of which is the inverse of the other. The basic method to find multiplicative inverses is Extended-Euclidean method. In this paper we will propose a new algorithm for computing the inverse, based on continues subtract fraction from integer and divide by fraction to obtain integer that will be used to compute the inverse d. The authors claim that the proposed method more efficient and faster than the existed methods.

Keywords - Multiplicative inverse, greater common divisor, Euclidean method, Stein method, Gordon method, Baghdad method

1. Introduction

Modular arithmetic plays an important role in cryptography. Many public-key schemes [2] involve modular exponentiation. Modular inversion, the computation of $b^{-1} \bmod a$ has a part in exponentiation based on addition-subtraction chains [6], as well as other applications in such public key systems.

The multiplicative inverse of e modulus n is an integer d such that $e*d \equiv 1 \bmod n$, d is called the inverse of e and denoted e^{-1} [5]. The study of inverse calculation was an intractable science due to lack of real improvement, the modulus inverse problem is a lot more difficult to solve [1]. However, there were only a few methods.

The first one is trivial and lengthy in calculating the inverse, because it is a sequential search. It starts by $d=1$, keep on adding 1 to d until $e*d \equiv 1 \bmod n$.

In [3] Euclidian described the algorithm in his book, Elements, written around 300 B.C. It is the oldest nontrivial algorithm that has survived to the present day, and it is still a good one. Euclid's algorithm is an efficient method to calculate the greatest common divisor of two integers without factoring them.

Euclidian algorithm can also compute the inverse of a number modulo n, sometimes this is called the extended Euclidean algorithm, this method is based on the idea that if $n > a$ then $\gcd(a,n) = \gcd(a, n \bmod a)$, also on finding $a*x + y*n = 1$ in which x is the multiplicative inverse.

Euclidian algorithm is approximately irrelevant to e or n, but other algorithms are affected by e and the modulus n.

2. Previous methods

In this section we will describe the methods that deal with the computing multiplicative inverse which are as follows:

2.1. Euclid algorithm

This method is based on the idea that if $n > e$ then $\gcd(e,n) = 1$, also on finding $e*x + y*n = 1$ in which x is the multiplicative inverse of e [4]. The algorithm is iterative and can be slow for large numbers. Knuth showed that the average number of divisions performed by the algorithm is $0.843*\log_2(n) + 1.47$ [2].

The method needs 8 variables, and used subtraction, multiplication, division, and

JOURNAL OF COMPUTING, VOLUME 1, ISSUE 1, DECEMBER 2009, ISSN: 2151-9617
HTTPS://SITES.GOOGLE.COM/SITE/JOURNALOFCOMPUTING/

comparison as operations, the complexity of $O(\log n)$.

Algorithm

Input: $e \in Z_n$ such that $\gcd(e,n)=1$.

Output: $e^{-1} \bmod n$ where $e^{-1}=i$ provided that it exists.

The algorithm is as follows:

1. Set $g \leftarrow n; u \leftarrow e; i \leftarrow 0; v \leftarrow 1;$
2. While $u>0$ do the following:
   $q \leftarrow \lfloor g/u \rfloor; t \leftarrow g - q*u;$
   $g \leftarrow u; u \leftarrow t; t \leftarrow i - q*v;$
   $i \leftarrow v; v \leftarrow t;$
3. If $i<0$ then
   $i \leftarrow n+i;$
4. $e^{-1} \leftarrow i$

Example

Let $e \leftarrow 7; n \leftarrow 60$

g	u	i	v	q	t
60	7	0	1	0	0
7	4	1	-8	8	-8
4	3	-8	9	1	9
3	1	9	-17	1	-17
1	0	-17	-52	3	-52

$e^{-1} \leftarrow n+i = 60+(-17) = 43$

2.2. Stein Method

In 1967, Stein introduced an inverse algorithm [7] and later improved by Penk Knuth. It is based on the observation that $\gcd(x,y) = \gcd(x/2,y)$ if x is even, also $\gcd(x,y)=2$, $\gcd(x/2,y/2)$ if both x,y are even, and $\gcd(x,y) = \gcd((x-y)/2,y)$ if x,y are both odd.

The algorithm needs about 11 variables, and uses addition, subtraction, multiplication, division and comparison, the complexity is $O(\log n)$.

Algorithm

Input: $e \in Z_n$ such that $\gcd(e,n)=1$.

Output: $e^{-1} \bmod n$ provided that it exists.

The algorithm is as follows:

While e and n is even do
$e \leftarrow \lfloor e/2 \rfloor; n \leftarrow \lfloor n/2 \rfloor;$
$u_1 \leftarrow 1; u_2 \leftarrow 0; u_3 \leftarrow e; v_1 \leftarrow n; v_2 \leftarrow 1-e; v_3 \leftarrow n;$ If e is odd then
$t_1 \leftarrow 0; t_2 \leftarrow -1; t_3 \leftarrow -n;$
Else $t_1 \leftarrow 1; t_2 \leftarrow 0; t_3 \leftarrow e;$
Repeat
While t_3 is even do
$t_3 \leftarrow \lfloor t_3/2 \rfloor;$
If t_1 and t_2 is even then
$t_1 \leftarrow \lfloor t_1/2 \rfloor; t_2 \leftarrow \lfloor t_2/2 \rfloor;$
Else $t_1 \leftarrow \lfloor (t_1+n)/2 \rfloor; t_2 \leftarrow \lfloor (t_2-e)/2 \rfloor;$
If $(t_3 > 0)$ then
$u_1 \leftarrow t_1; u_2 \leftarrow t_2; u_3 \leftarrow t_3;$
Else $v_1 \leftarrow n-t_1; v_2 \leftarrow -(e+t_2); v_3 \leftarrow -t_3;$
$t_1 \leftarrow u_1 - v_1; t_2 \leftarrow u_2 - v_2; t_3 \leftarrow u_3 - v_3;$
If $(t_1 < 0)$ then
$t_1 \leftarrow t_1 + n; t_2 \leftarrow t_2 - e;$
Until $t_3 = 0;$
$e^{-1} \leftarrow u_1;$

Example

Let $e=7; n=60;$

e	n	u_1	u_2	u_3	v_1	v_2	v_3	t_1	t_2	t_3
7	60	1	0	7	60	-6	60	0	-1	-60
								30	-4	-30
								15	-2	-15
					45	-5	15			
								-44	5	-8
								16	-2	
								8	-1	-4
								34	-4	-2
								17	-2	-1
					43		1			
								-42	5	6
								-18	-2	
								9	-1	3
		9	-1	3						
								-43	4	2
								-26	-3	
								43	-5	1
		43	-5	1						
								0	0	0

$e^{-1} \leftarrow u_1 = 43$

2.3. Gordon Method

In 1989, Gordon [2] described another algorithm for computing an inverse. It is based on the observation that q at Euclidian method does not need to be the remainder of n/a but it can be any power of 2 up to that limit [4]. The algorithm needs about 9 variables, and uses addition, subtraction, comparison, and shifting. The complexity of the algorithm is $O(\log n)$

Algorithm

Input: $e \in Z_n$ such that $\gcd(e,n)=1$.

Output: $e^{-1} \bmod n$ provided that it exists.

The algorithm is as follows:

$g \leftarrow n; i \leftarrow 0; v \leftarrow 1; u \leftarrow e;$
Repeat
$s \leftarrow -1; p \leftarrow 0;$
If $u > g$ then

JOURNAL OF COMPUTING, VOLUME 1, ISSUE 1, DECEMBER 2009, ISSN: 2151-9617
HTTPS://SITES.GOOGLE.COM/SITE/JOURNALOFCOMPUTING/

$t \leftarrow 0;$
Else
$p \leftarrow 1; t \leftarrow u;$
While $(t \le g)$ do
$s \leftarrow s+1;$
$t \leftarrow$ Left shift t by 1;
$t \leftarrow$ Right shift t by 1;
$t \leftarrow g-t; g \leftarrow u; u \leftarrow t; t \leftarrow i; i \leftarrow v;$
If $p = 1$ then
$v \leftarrow$ Left shift v by $s;$
$t \leftarrow t-v;$
$v \leftarrow t;$
Until $u = 0$ or $u = g;$
If $i < 0$ then
$i \leftarrow n+i;$
$e^{-1} \leftarrow i;$

Example

Let $e \leftarrow 7; n \leftarrow 60$

g	u	i	v	s	p	t
60	7	0	1	0	1	14
				1		28
				2		58
				3		112
						56
7	4					4
		1				0
			8			−8
				−1	0	
					1	4
				0		8
4	3					3
		-8				1
						9
			9			
				−1	0	
					1	3
				0		6
						3
1	1					1
		9				−8
						−17
			−17	−1	0	
					1	1
				0		2
				1		4
						2
1						1
	−17					9
			−3			43
			43			

$e^{-1} \leftarrow 60-17 = 43$

2.4. Baghdad algorithm

In 2004, Sattar Aboud [6] introduced another algorithm entitled "Baghdad method" to calculate the inverse. The idea behind Baghdad method is very simple involving adding 1 to the modulus n and then divides the result by the exponent e. Then keep on adding the result to the modulus n and divide the new result by the exponent e until an integer is obtain.

The algorithm needs only 5 variables, and uses addition and division only. The complexity of the algorithm is $O(\log n)$

Algorithm

Input: $e \in Z_n$ such that $\gcd(e,n) = 1$

Output: $e^{-1} \bmod n$ provided that it exists

The algorithm is as follows:

Set $d \leftarrow 1;$
Repeat
$d = (d+n)/e;$
Until d is integer
$e^{-1} \leftarrow d;$

Example

Let $e \leftarrow 7; n \leftarrow 60;$

d	*result*
(1+60)/7	not integer
(61+60)/7	not integer
(121+60)/7	not integer
(181+60)/7	not integer
(241+60)/7	integer match

$e^{-1} \leftarrow d = 43$

3. Fast Fraction-Integer Method

The idea behind the proposal method is a very simple, based on continues subtract fraction from integer and divide by fraction to obtain integer that will be used to compute the inverse d. The algorithm needs only 6 variables, and uses addition and division only. The complexity of the algorithm is $O(\log n)$

Algorithm

Input: $e \in Z_n$ such that $\gcd(e,n) = 1$

Output: $e^{-1} \bmod n$ provided that it exists.

The algorithm as follows:

$r : real;$
$i = 1;$
$s_f = (n+1 \bmod e)/e;$
$d_f = (n \bmod e)/e;$
If $s_f = 0$ then
Stop;
Else
Repeat
$r = ((i - s_f)/d_f);$

JOURNAL OF COMPUTING, VOLUME 1, ISSUE 1, DECEMBER 2009, ISSN: 2151-9617
HTTPS://SITES.GOOGLE.COM/SITE/JOURNALOFCOMPUTING/

$i = i + 1;$
Until r is integer
$d = (n * (r + 1)) + 1 / e;$

Example
Let $e \leftarrow 7; n \leftarrow 60;$

i	s_f	d_f	r
1	0.71428	0.57142	0.50001
2			2.25004
3			4.00000

$d = (60 * (r + 1)) + 1 / e$
$= (60*(4+1)) + 1/7$
$= (60*5) + 1/7$
$= 301/7$
$= 43$

3.1 Proof of Fast Fraction-Integer Method

In order to prove the algorithm, we need to prove that the algorithm will give integer number only when d is the inverse of e. As we know that if d is the inverse of e then

1. Both e, d are positive integer numbers between $[1..n]$ .... (1)
2. $\gcd(e, n) = 1$ .....(2)
3. $e * d \equiv 1 \bmod n$, it means that $e * d = 1 + k * n$ for $k \in Z$ .....(3)

So $d = (1 + k * n) / e$
$= 1/e + k * n / e$ .....(4)

From the algorithm of Fast Fraction-Integer Method we see that $d = (n * (r + 1)) + 1 / e;$ this will repeated i times until d ..... (5)

From that we know that the algorithm above is correct for $i = k$, but if this is the case we need to prove that (5) will give none integer for all values of $i < k$, and the only integer value is when $i = k$, so we know d is an integer so $(1 + k * n) / e$ is also integer for integer value of k.

Then we need to proof that $(1 + i * n) / e$ is never an integer for all values of i between $[1, k - 1]$. Assume that there is another value of i where $i < i < k$ such that $d = (1 + i * n) / e$ is also an integer, it means that $i = k - 1$ ---------- (6)

Then $d = (1 + (k - 1) * n) / e$ will be integer. So
$d = (1 + k * n - n) / e$
$= (1 + k * n) / e - n / e$
$= 1/e + k * n / e - n / e$

But by $1/e + k * n / e$ is integer, and by that $\gcd(e, n)$ should be 1. So if there is no greater common divisor between e and n except 1, that means n / e is a non integer value.

Thus subtracting a non integer value form an integer value will yield d is not an integer. This will contradict our assumption (that d is an integer).

Now assume that there exist $i = k - q$ such that d is an integer for q between $[1, k - 1]$. Then
$d = (1 + (k - q) * n) / e$
$= 1/e + k * n / e - q * n / e$

If this to be integer then $q * n / e$ must be integer, but since $\gcd(e, n) = 1$ then q must be a multiple of e so $d = 1/e + k * n / e - x * n$ (5)

This will lead to d being a negative number $d < 0$ but from definition we know that both e and d must be positive (1) so there is no values for x that satisfy the definition. So the only value for q that satisfy the conditions is when $q = 0$ and that $i = k$.

3.2Problem of Fast Fraction-Integer method

We have proved that Fast Fraction-Integer algorithm is correct, but the question is that is it implemental? Yes the algorithm will terminate giving the correct answer when implemented using the computer programming languages.

Let dm be the mathematical value of d where $d = dm$. Let dc be the calculated value of d in the computer memory and registers. Let ξ be the error in calculating, between the mathematical value and the computer value (round off error).
So $dm = (1m + km * nm) / em$
$= 1m / em + km * nm / em$
$= (1 / e)m + (k * n / e)m$

But we know that the calculated value of fractions is never exactly as the mathematical value for big values of e that when used to divide 1 and n will give a cyclic fraction number.

So $(1 / e)m = (1 / e)c + \xi 1$ and $(n / e)m = (n / e)c + \xi 2$ where $\xi 1 << (1 / e)c$, $\xi 2 << (n / e)c$ and $dc = (1 / e)c + \xi 1 + k * \xi 2$ such errors will yield that either $dm \le dc$ or $dm \ge dc$, $dm - dc$ if and only if $\xi 1 + k * \xi 2 = 0$ it means that $(1 / e)m = (1 / e)c$, $(n / e)m = (n / e)c$ We know that the error $\xi 1, \xi 2$ is small, but multiplying $\xi 2$ with k will give big value to the error and the

JOURNAL OF COMPUTING, VOLUME 1, ISSUE 1, DECEMBER 2009, ISSN: 2151-9617
HTTPS://SITES.GOOGLE.COM/SITE/JOURNALOFCOMPUTING/

error will multiply by k, so as k is increasing the error also will increase so the best approach is to use small values for e.

4. Conclusions

For security reasons, cryptography recommends smaller values for public keys and bigger values for private keys [4]. The suggested algorithm needs lower values for public keys (lower value of e) and higher values for private key, which is fully compatible with the preferred cryptography algorithm. The method is simple, fast and needs less storage, and its complexity is also less.

References

1. B. Schneier, applied Cryptography, Second Edition, John Wiley and sons, 1996, p 246.
2. J. Gordon, Fast Multiplicative inverse in modular arithmetic, Cryptography and Coding, Clarendon Press Oxford, 1989, .pp 269 - 279.
3. D. E. Knuth, The art of computer programming, 2nd Ed., Addison - Wesley, Vol. 2, 1981, pp 319, 321, 339, 599.
4. A. Menezes. et al, Handbook of applied cryptography, CRT Press, 1996, p 67, p 71.
5. R. Rivest, A. Shamir., and L. Adlemen, A method for obtaining digital signatures and public key cryptosystems, ACM, 1978, pp 120-126.
6. Sattar J Aboud, Baghdad Method for calculating Multiplicative Inverse, The International Conference on Information Technology (ITCC 2004), IEEE, 5-7 April 2004, Las Vegas, U .S. A
7. J. Stein, Comp. Phys, 1, (1967), p 397-405.

JOURNAL OF COMPUTING, VOLUME 3, ISSUE 2, FEBRUARY 2011, ISSN 2151-9617
HTTPS://SITES.GOOGLE.COM/SITE/JOURNALOFCOMPUTING/
WWW.JOURNALOFCOMPUTING.ORG

Anonymous and Non-Repudiation E-Cash Scheme with Partially Blind Signature

Hani M. AL-Matari[1], Abdalnaseer A. Hajer[1] and Nidal F. Shilbayeh[1]
[1]Middle East University for Graduate Studies, Faculty of IT, Jordan-Amman

Abstract— Partially Blind Signature techniques played an important role in building e-cash systems. It allows the signer to include pre-agreed information such as expiration date or collateral conditions in the resulting signature. In this paper, we proposed a non-repudiation and anonymous e-cash scheme based on partially blind signature that enables the Judge to specify a dishonest customer, bank, or blind office. In addition to that, our scheme is considered as a multi-purpose scheme because it satisfies the integration of multi spendable and divisible coins. We also analyze the efficiency and the security of the proposed scheme.

Index Terms— e-cash scheme, blind signature scheme, partially blind signature scheme, hash function, RSA scheme, Elgamal scheme.

1 Introduction

THE concept of electronic cash was proposed by [1] in 1982. The main idea is that, even though the same party (a bank) is responsible for giving out electronic coins, and for later accepting them for deposit, the withdrawal and the spending protocols are designed in such a way that it is impossible to identify when a particular coin was spent. I.e., the withdrawal protocol does not reveal any information to the bank that would later enable it to trace how a coin was spent.

In a blind signature, the customer requests the signer to sign on a blinded data. The client then derives the wanted signature from the signed blind data. When the customer finally hands in the message and its signature, the signer is able to verify this signature, but is unable to link this signed message to the previous signing process instance.
Chaum's scheme is based on RSA public key cryptosystem [11] and its security depends on the difficulty of integer factorization. Blind signature schemes see a great deal of use in applications where sender privacy is important. This includes various "digital cash" schemes and voting protocols. In fact this anonymity might be misused by fraudulent to perform a perfect crime [13]. For instance stealing of the private keys, money laundry, and blackmailing of coins. The use of blindfolded protocols in the banks is considered as modern threats [5]. These threats are just prevented in the scheme of [15], which is also not efficient as it needs the trusted authority interaction in e-payment schemes.
There are numerous papers are suggested employing blind signature schemes to design an e-payment protocols, which satisfies the needs of both the banks and the entities [4],[19],[20].
The problem with the current protocols is that difficulty resultants from these trust relations when customer can repudiate his bad activity [14].
The system [2] is based on blind signatures. In this model blinding is sub contracted to a trustee, using identity-linked pseudonyms (PIDs) to achieve anonymity. The bank acts as a Certification Authority (CA) that provides the user the certified PIDs (therefore the bank can link the pseudonym to the real identity of the user). The trustee is designed as a Blinding Office, it can link all payments made under the same PID. The Customer may use the PIDs to pseudo-identify himself to the BlindOffce.

The notion of partially blind signatures was introduced in [6].Their construction, based on RSA. In a partially blind signature scheme the signer can impose the common information, for example date, on the signature such that the verifier needs the message, the common information and the signature to check the validity of this signature [7]. In their scheme, the bank is clearly notified the common information- the expiration date of an e-cash. With the partially blind signature, the bank assures that the signed e-cash carry the agreed common information-expiration date. With this common information, the bank needs only to keep the still-alive e-cashes in the database to prevent double spending. Those expired e-cashes could be eliminated from the database without any trouble. This partial blindness property preserves the unlinkability of the blind signature, but imposes the common information on the signature.

Based on the discrete logarithm problem, [7] proposed

- *Hani M. Al-Matari is doing a master degree in computer science at Middle East University, Amman, Jordan.*
- *AbdAlnasser A. Hagar is doing a master degree in computer information systems at Middle East University, Amman, Jordan.*
- *Nidal F. Shilbayeh is with the Computer Science Department, Middle East University, Amman, Jordan.*

JOURNAL OF COMPUTING, VOLUME 3, ISSUE 2, FEBRUARY 2011, ISSN 2151-9617
HTTPS://SITES.GOOGLE.COM/SITE/JOURNALOFCOMPUTING/
WWW.JOURNALOFCOMPUTING.ORG

a partially blind signature, and proposed an efficient E-cash system. Later, in [18], based on the discrete logarithm problem, proposed a partially blind (t, n) threshold signature scheme in which any t out of n signers in a group can represent the group to sign the partially blind threshold signature.

In [9] propose a secure partially blind signature scheme based on factoring and discrete logarithms and show that the proposed scheme satisfies the partial blindness, randomization, unlinkability and unforgeability properties.

This paper is organized as follows. In section 2 we explain the general e-cash scheme, which is applicable for both the M'Raihi scheme and prposed scheme. In section 3 cryptography requirement will be discussed. In section 4 we described the propose scheme. Whilst the section 5 illustrate the security of the propose scheme. Finally we state concluding remarks in section 6.

2 E-CASH SCHEME

In this section we explain the general e-cash scheme, which is applicable for both the M'Raihi scheme and prposed scheme.
This e-cash scheme consists of five entities, Customer, Bank, BlindOffice, Payee and judge.

- **Customer:** purchases goods or services from payee using e-cash
- **Bank:** issues e-cash and maintains bank account for customers and payee.
- **BlindOffice:** pseudo identity escrow agency, it can link all payments made under the same PID
- **Payee:** sells goods or services to customer, and deposits e-cash to bank
- **Judge:** to adjudicate between the four entities; Bank, Customer, BlindOffice and payee.

The bank acts as a Certification Authority that provides the customer the certified PIDs (therefore the bank can link the pseudonym to the real identity of the customer). The trustee is designed as a BlindingOffice, it can link all payments made under the same PID. The customer may use the PIDs to pseudo-identify himself to the BlindOffice.
The e-cash scheme can be acts as follows. Customer obtains a coin blindly signed by Bank. The Bank holds a relation prove between customer's true identifier (ID) and pseudo identifier (PID), BlindOffice participated in the blind signature, preserves another relation prove among PID and Coin. To spend Coin, Customer proofs to Payee that he has knew of secret key x accordance to Coin. If Coin is misused, for example over spend, Bank and BlindOffice will cooperate to make a link amongst ID and Coin, Judge will be participated in these tracing steps to judge.
Bank has the exponent key type signatures denoted by (S_{bank}, V_{bank}), such that V_{bank} is known to customer, BlindOffice and payee.
Customer also has the exponent key type signature denoted by (S_{cust}, V_{cust}), such that V_{cust} is known to bank.

Assume that BlindOffice has an exponent public-key denoted by (E_{BO}, D_{BO}), such that E_{BO} is known to customer

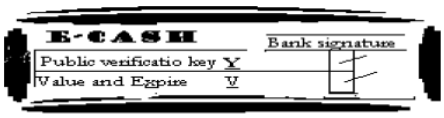


Fig 1, Coin fields

and bank. Judge could be verified all schemes.
The coin contains three fields as in figure 1.

1. The public verification key denoted by y for a public key type signature scheme. The corresponding secret signature key is represented by x.
2. The data item v (pre-define message between Bank and Customer) having some pertinent data concerning Coin, for example its value and expiry dates.
3. The bank's digital signature on both y and v.

In fact, we need the signature scheme of the bank to have some particular property, that is $S_{Bank}(m_2) = S_{Bank}(m_1 . m_2)$, which holds for RSA. Certainly this is usually a largely unfavorable feature for a signature scheme, and is one reason why RSA must always be employed in combination with a special redundancy function or a one-way hash function [3] [8]. In our instance, we either clearly state the employ of a one-way hash function indicated by h(m) for instance message (m) with S_{Bank}, or prevent deceptions resultant from the use of direct RSA by other ways. We indicate a blinding function by BF, and allow it be an inverse of the signature scheme, so that $S_{Bank}(BF(m_1)m_2) = m_1 S_{Bank}(m_2)$ for each m_1, m_2. If Bank's signature scheme is RSA, then BF is just exponentiation employing the public verification key.

3 CRYPTOGRAPHY REQIURMENT

An ideal e-cash system must satisfy the following properties:

- Unforgeability: inability to forge the valid e-cash.
- Anonymity: anyone cannot trace e-cash owner and cannot know what the customer bought.
- Anonymous revocation: legal coin or owner tracing is possible to prevent crimes.
- Double spending prevention: the same e-cash must not allow spending twice.
- Off-line: when a customer gives e-cash to a payee, it is not need to connect to the bank on-line.
- Transferability: when a customer receives an e-cash in a transaction, he may spend it without depositing the coin first and getting a new e-cash issued from bank.

3.1 On-line and double spending prevention

An issue with the basic scheme is that the Customer may behave dishonestly and try to spend a e-coin more than once. The Issuer will discover this when recording the corresponding Blank for a second time in its database of spent e-coins but the Payee on its own cannot check

JOURNAL OF COMPUTING, VOLUME 3, ISSUE 2, FEBRUARY 2011, ISSN 2151-9617
HTTPS://SITES.GOOGLE.COM/SITE/JOURNALOFCOMPUTING/
WWW.JOURNALOFCOMPUTING.ORG

that a e-cash has not been spent before. Thus the Issuer needs to be online to provide the Payee with an assurance, before the Payee completes the transaction with the Customer, that the e-coin he has been offered has not already been spent

3.2 Off –line and tracing double Spending

An alternative to having the Issuer online to prevent double spending is to have a means of tracing Customer who double spend and to take action against them. This requires that the Customer's identity is encoded in the Blank in such a way that is only revealed if the User double spends.

4 PROPSED SCHEME

Our electronic cash scheme consists of five entities, Customer, Bank, BlindOffice, Payee and judge.see figure 2. Our scheme is works as follows:
Customer obtains a coin signed partially blind by Bank. Then Bank holds a relation proof among customer real

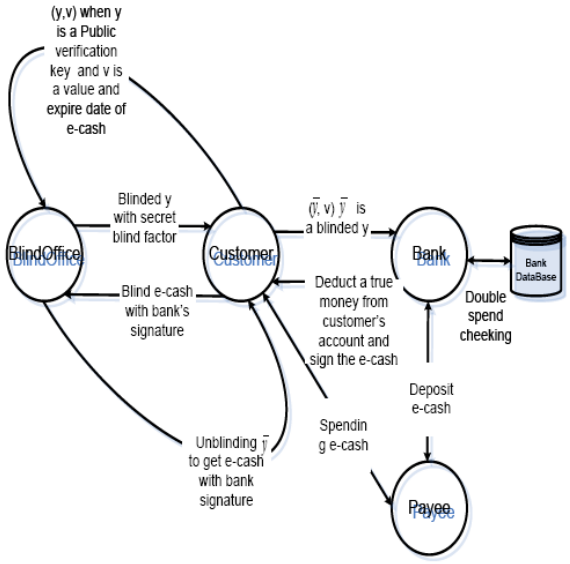


Fig 2, proposed scheme

identifier ID and pseudo identifier *PID*.
BlindOffice participated on a partially blind signature, holds another relation proof among customer pseudo identifier PID and coins.
To spend coins, customer need to proofs to payee that he know x.
If coin is misused, for example double spending, bank and BlindOffice will work together to construct a link among *ID* and coin, and judge will be participated in this procedure to judge.

Previous scenario will be achieved via three protocols as follows

4.1 The initial Protocol

The bank publishes: (*n*, *e*, *f*), which *n* and *e* is a standard RSA public key and the private key *d* is the multiplicative inverse of *e*.
And *f* is an appropriate public exponent generating function. With RSA based partially blind signature scenario $f()$ a public exponent function satisfying $f(v)$ must be different for different values of *v*; where *v* is a message predefined by the bank and contains an expiration date of the e-cash and BlindOffice's name. $f(v)$ must be relatively prime to (*n*) and Each output has at least one unique prime factor.
The Customer and *Bank generate* a shared secret *s*.Then *Bank* signs a one way hash function of *s*, namely $S_{BANK}(h(s))$ which is employed to build *PID* by concatenating it with $E_{BO}(s)$.

4.2 The Withdraw Protocol

The withdraw protocol can be dividing to three phases: the first phases called the binding phase as in figure 3.
If a customer decides to withdraw e-cash from the bank,as in figure 3; he/she randomly chooses x and calculate y then sends $E_{BO}(y,v)$ to BlindOffice where v is a message predefine by the bank and contain an expiration date of the e-cash, value, and BlidOffice's name.
The BlidOffice choose blind factor r and calculate $\bar{y}$ =BF (r, $f(v)$).y and passes $\bar{y}$ to customer.
In the seconde phase which called signing phase (figure 4) the customer then sends $\bar{y}$ and v to bank. The bank

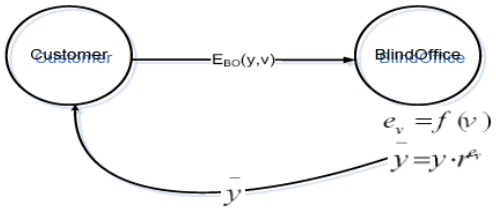


Fig 3, Withdrawal protocol (blinding phase)

verifies whether or not v is correct; If it is correct the bank calculates t= E_{BO} (S_B ($\bar{y}$)) and send t to customer and deduct true coins from customer's account and keep a relation proofs {ID, $\bar{y}$ }.
Customer passes t to BlindOffice. Then BlindOffice recover $\bar{t}$ to get $S_B(\bar{y})$ then unblinds $S_B(y)$ then build coins and passes it to customer and holds a relation {PID, Coins} this

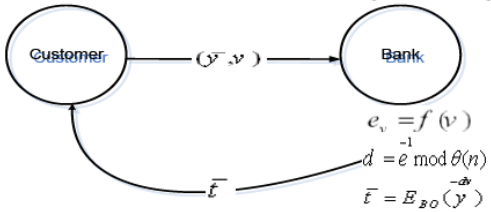


Fig 4, Withdrawal protocol (signing phase)

phase is called unblinding phase figure 5.

JOURNAL OF COMPUTING, VOLUME 3, ISSUE 2, FEBRUARY 2011, ISSN 2151-9617
HTTPS://SITES.GOOGLE.COM/SITE/JOURNALOFCOMPUTING/
WWW.JOURNALOFCOMPUTING.ORG

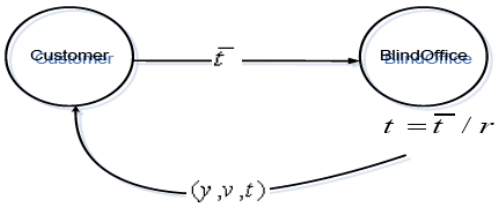


Fig 5, Withdrawal protocol (unblinding e-cash phase)

4.3 The Spending Protocol

As in figure 6: when customer use the coins she/he passes it to payee then payee create a challenge message to prove customer knows x.
Payee claims a true coin back from Bank later

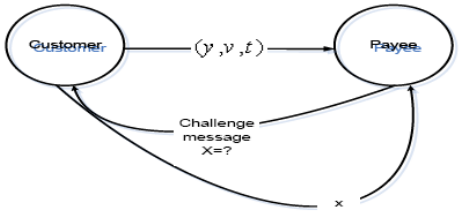


Fig 5, Spinding protocol

4.4 Double Spending occur

Assume the coin is double spending, Bank will request a tracing steps in which Bank and BlindOffice work together to construct a link among C and ID , relied on {ID, PID} and {PID,Coin}.

5 CORRECTNESS

5.1 Partially blinding phase:

$e_v = f(v)$

$\bar{y} = y \cdot r^{e_v} \bmod n$
$e = f(v)$
$d = \bar{e}^{-1} \bmod \theta(n)$
$t = \bar{t} / r = E_{BO}(\bar{y}^{-dv})$
(y, v, t)
$challenge$

5.2 Signing Phase:

$d_v = \bar{e}_v^{-1} \bmod \theta(n) \cdot$

$\bar{t} = \bar{y}^{-dv} = (y \cdot r^{e_v})^{d_v} = y^{d_v} \cdot r \bmod(n)$

5.3 Unblinding Phase

$t = \bar{t} / r = (y^{d_v} \cdot r) / r = y^{d_v} \bmod n.$

5.4 Verification Phase

Theorem: If (v, y, t) is a signature of the message y produced by the proposed scheme, then $y \equiv t^{e_v} \bmod n$
Proof:

$$t^{e_v} \stackrel{?}{\equiv} y \bmod n$$
$$t^{e_v} \equiv (r^{-1} \cdot \bar{t})^{e_v}$$
$$\equiv (r^{-1} \cdot y^{-d_v})^{e_v}$$
$$\equiv (r^{-1} \cdot (r^{e_v} \cdot y)^{d_v})^{e_v}$$
$$\equiv (r^{-1} \cdot r \cdot y^{d_v})^{e_v}$$
$$\equiv (y^{d_v})^{e_v}$$
$$\equiv y \bmod n$$

6 SCUIRTY OF THE SCHEME

In our scheme we claim that customer, BlindOffice and bank can't repudiate any steps done by him also the entities in our scheme can not impersonate each other, the other important benefit that the bank needs only to keep the still-alive e-cashes in the database to prevent double spending.
We now plan the proofs that the our scheme owns this security abilities

Theorem 1: Customer can not get Coin without the participation of Bank and BlindOffice.
Proof 1: to get Coin without Bank or BlindOffice be included, Customer should be capable to calculate s from r or E_{BO} (t), each of which is supposed to be infeasible.

Theorem 2: Neither Customer or BlindOffice can change value of v after customer sends it to BlindOffice.
Proof 2: To change value of v to $v1$ BlindOffice need to calculate S_{cust} on $v1$, and if customer change value of v to v1and send it to bank, the bank will be reject this value when they verify is v correct or not according to $\bar{y}$.

Theorem 3: The member who is the publisher of x can just spend a valid coin relevant to a secret key x.
Proof 3: Suppose that given y and further associated public data, it is computationally infeasible to decrypt x, x is given only to its publisher and is not exposed to any other person. So, because knowing of x is needed to spend coin, the finding follows.

Theorem 4: The BlindOffice cannot impersonate a customer to Bank.
Proof 4: To impersonate Customer to Bank, Blind Office should find Customer's digital signature on the $\bar{y}$ and v, which is supposed to be infeasible.

Theorem 5: When Bank impersonates a customer to get and to spend a coin; he cannot say that Customer published the coin.
Proof 5: To verify {ID, $\bar{y}$ }, Bank wants Customer's signature on y and v. Such a signature cannot be found, in spite

JOURNAL OF COMPUTING, VOLUME 3, ISSUE 2, FEBRUARY 2011, ISSN 2151-9617
HTTPS://SITES.GOOGLE.COM/SITE/JOURNALOFCOMPUTING/
WWW.JOURNALOFCOMPUTING.ORG

of BlindOffice's cooperation.

7 Conclusion

Our proposed e-cash scheme with partially blind signature designed to provide characteristics of anonymity, non-repudiation and traceability. The proposed scheme is suitable for on-line; the bank assures that the signed e-cash carry the agreed common information- expiration date. With this common information, the bank needs only to keep the still-alive e-cashes in the database to prevent double spending. The proposed scheme is multi-purpose, as it allows the integration of multi spendable and divisible coins.

References

[1] Chaum, D., *Blind Signature for Untraceable Payments*, Advances in Cryptology. Crypto'82. Plenum Press.1983.

[2] D. M'Raihi, Cost Effective Payment Schemes with Piracy Regulation, Advances in Cryptology-ASIACRYPT 96, Lecture Notes in Computer Science 1163, pages 266-275, Springer-Verlag, 1996.

[3] D. R. Stinson."*Cryptography, Theory and Practice*". CRC Press, Boca Raton, 1995.

[4] Kungpisdan, S., B. Srivnivasan and P. Le. *A Secure Account-Based Mobile Payment Protocol*. Proceedings of the Intl. Conference on Information Technology: Coding and Computing, (ITCC'04). 2004.

[5] Liu, J.,K. Wei and S. Wong, *Recoverable and Untraceable E-Cash*, EUROCON'2001: Trends in Communications. Intl. Conference on Information Technology, Vol. (1): 342-349. 2001.

[6] M. Abe and E. Fujisaki. *How to date blind signatures*. In K. Kim and T. Mat- sumoto, editors, Advances in Cryptology { ASIACRYPT '96, volume 1163 of Lec- ture Notes in Computer Science, 1996.

[7] M. Abe and J. Camenisch. *Partially blind signatures schemes for the DSS and for a discrete log*. Symposium on Cryptography and Information Security, 1997.

[8] Menezes, P.C. van Oorschot, and S.A. Vanstone, *Handbook of applied cryptography*, 1996.

[9] N.M.F. Tahat, S.M.A. Shatnawi and E.S. Ismail, A New Partially Blind Signature Based on Factoring and Discrete Logarithms

[10] R.A. Mollin, Codes — The *Guide to Secrecy from Ancient to Modern Times*, Chapman and Hall/CRC (2005).

[11] R.L. Rivest, A. Shamir, and L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Communications of the A.C.M.21 (1978).

[12] S. J. Aboud and M. A. AL-Fayoumi. *Anonymous and Non-Repudiation E-Payment Protocol*. American Journal of Applied Sciences 4 (8): 538-542, 2007.

[13] Solms,V and D. Naccache, *Blind Signatures and Perfect Crimes*. Intl. J. Computers& Security, Vol. (11): 581-583. 1992.

[14] Song, R and L. Korba. *How to Make E-cash with Non-Repudiation and Anonymity*. Proceedings of the Intl. Conference on Information Technology: Coding and Computing, (ITCC'04):167-172. 2004.

[15] Susan Hohenberger, Advances *in Signatures Encryption and E-Cash from Bilinear Group*. PhDThesis. MIT. 2006.

[16] T. ElGamal, *A public key cryptosystem and signature scheme based on discrete logarithms*, IEEE Transactions on Information Theory.1985.

[17] W. Diffie and M.E. Hellman, *New directions in cryptography*, IEEE Transactions on Information Theory 22 (1976).

[18] W.S. Juang, and C.L. Lei *Partially blind threshold signatures based on discrete logarithm*. Computer Communications, 1999.

[19] Yang, L and J. Li. *Application Study on Public Key Cryptography in Mobile Payment*, Proceeding of the 5th WSEAS Intl. Conference on Information Security and Privacy, Venice. 2006.

[20] Zheng, X and D. Chen. *Study of Mobile Payments System Proceedings of the IEEE Intl.* Conference on E-Commerce, (CEC'03). , 2003

Mr. Hani M. AL-Matari received his B.Sc. degree in Faculty of IT, Applied Science University, Jordan in 2007. He is currently doing Master degree in computer science at Middle East University. His research interests include cryptography and e-payment systems.

Mr. AbdAlnasser A. Hagar received his B.Sc. degree in Faculty of IT, Applied Science University, Jordan in 2008. He is currently doing Master degree in computer information systems at Middle East University. His research interests include cryptography and Building Database systems.

Prof. Nidal F. Shilbayeh received the B.Sc. degree in computer science from Yarmouk University, Irbid, Jordan in 1988, the MS degree in computer science from Montclair State University, New Jersey, USA in 1992, and the PhD in computer science from Rajesthan University, Rajesthan, India in 1997. He is currently a professor at Middle East University. His research interests include digit recognition, pattern recognition, face recognition, security, nose systems, watermarking, and biometrics.