A BLIND SIGNATURE SCHEME USING BIOMETRIC FEATURE VALUE

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology
In
Computer Science and Engineering

By

JYOTI PRAKASH MUDULI
ROLL NO: 10606032
SANJAYA KUMAR PARIDA
ROLL NO: 10606035

Department of Computer Science and Engineering
National Institute of Technology
Rourkela
2010

A BLIND SIGNATURE SCHEME USING BIOMETRIC FEATURE VALUE

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology
In
Computer Science and Engineering

By

JYOTI PRAKASH MUDULI
ROLL NO: 10606032
SANJAYA KUMAR PARIDA
ROLL NO: 10606035

Under the Guidance of
Prof. Sujata Mohanty

Department of Computer Science and Engineering
National Institute of Technology
Rourkela
2010

National Institute of Technology
Rourkela

CERTIFICATE

This is to certify that the thesis entitled, "**A BLIND SIGNATURE SCHEME USING BIOMETRIC FEATURE VALUE**" submitted by **Jyoti Prakash Muduli, Roll No: 10606032 and Sanjaya Kumar Parida, Roll No: 10606035** in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in **Computer Science and Engineering** at the National Institute of Technology, Rourkela is an authentic work carried out by them under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other university / institute for the award of any Degree or Diploma.

Date: 7th May, 2010

Prof. Sujata Mohanty
Dept. of Computer Science and Engineering
National Institute of Technology
Rourkela - 769008

ACKNOWLEDGEMENT

We avail this opportunity to extend our hearty indebtedness to our guide **Prof. Sujata Mohanty**, Computer Science Engineering Department, for their valuable guidance, constant encouragement and kind help at different stages for the execution of this dissertation work.

We also express our sincere gratitude to **Prof. B. Majhi**, Head of the Department, Computer Science Engineering, for providing valuable departmental facilities.

Submitted By:

Jyoti Prakash Muduli
Roll No: 10606032
Computer Science Engineering
National Institute of Technology
Rourkela

Sanjaya Kumar Parida
Roll No: 10606035
Computer Science Engineering
National Institute of Technology
Rourkela

ABSTRACT

Blind signature has been one of the most charming research fields of public key cryptography through which authenticity, data integrity and non-repudiation can be verified. Our research is based on the blind signature schemes which are based on two hard problems – Integer factorization and discrete logarithm problems. Here biological information like finger prints, iris, retina DNA, tissue and other features whatever its kind which are unique to an individual are embedded into private key and generate cryptographic key which consists of private and public key in the public key cryptosystem. Since biological information is personal identification data, it should be positioned as a personal secret key for a system. In this schemes an attacker intends to reveal the private key knowing the public key, has to solve both the hard problems i.e. for the private key which is a part of the cryptographic key and the biological information incorporated in it. We have to generate a cryptographic key using biometric data which is called biometric cryptographic key and also using that key to put signature on a document. Then using the signature we have to verify the authenticity and integrity of the original message. The verification of the message ensures the security involved in the scheme due to use of complex mathematical equations like modular arithmetic and quadratic residue as well.

CONTENTS

List of Figures

Chapter1

Introduction

1.1 What is a Digital signature?

In computing, the method of using encryption is to certify the source and integrity of a particular electronic document. Because all ASCII characters look the same no matter who types those, methods have to be found to certify the origins of particular messages if they are to be legally binding for electronic commerce or other transactions. One type of digital signature commonly seen on the internet is generated by the program Pretty Good Privacy (PGP), which adds a digest of the message to the signature. Digital signatures play an essential part in authenticating electronic commerce transactions.

A digital signature or digital signature scheme is a complex mathematical scheme for demonstrating the authenticity of a digital message or document. A valid digital signature gives a recipient reason to believe that the message was created by a known sender, and that it was not altered in transit. Digital signatures are commonly used for software distribution, financial transactions and defence system and in other cases where it is important to detect forgery and tampering.

Digital signatures are often used to implement electronic signatures, a broader term that refers to any electronic data that carries the intent of a signature, but not all electronic signatures use digital signatures. Digital signatures employ a type of asymmetric 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. Digital signatures are equivalent to traditional handwritten signatures in many respects; properly implemented digital signatures are more difficult to forge than the handwritten type. Digital signature schemes in the sense used here are cryptographically based, and must be implemented properly to be effective. Digital signatures can also provide non-repudiation, meaning that the signer cannot successfully claim they did not sign a message, while also claiming their private key remains secret; further, some non-repudiation schemes offer a time stamp for the digital signature, so that even if the private key is exposed, the signature is valid nonetheless[11]. Digitally signed messages may be

anything representable as a bit string: examples include electronic mail, contracts, or a message sent via some other cryptographic protocol.

A digital signature scheme typically consists of three algorithms:

1. A key generation algorithm that selects a private key uniformly at random from a set of possible private keys. The algorithm outputs the private key and a corresponding public key.
2. A signing algorithm which, given a message and a private key, produces a signature.
3. A signature verifying algorithm which given a message, public key and a signature, either accepts or rejects the message's claim to authenticity.[7,11]

But authentication is the utmost priority in the proposed project. In situations where there is not complete trust between sender and receiver, something more than authentication is needed. This opens the way the introduction of digital signature in the scheme. Digital signature must have the following properties.

- It must verify the author and the date and time of the signature.
- It must to authenticate the contents at the time of the signature.
- It must be verifiable by third parties, to resolve disputes

On the basis of these properties, we can formulate the following things which are required for a digital signature scheme:

- The signature must be a bit pattern that depends on the message being signed by anyone.
- The signature must use some information unique to the sender, to prevent both forgery and denial.
- It must be relatively easier to produce the digital signature.
- It must be relatively easy to recognize and verify the digital signature.

It must be computationally infeasible to forge a digital signature, either by constructing a new message for an existing digital signature or by constructing a fraudulent digital signature for a given message. It must be practical to retain a copy of the digital signature in storage.[11]

A variety of approaches has been proposed for the digital signature function. These approaches fall into two categories:

1. Direct Digital Signature
2. Arbitrated Digital Signature

1.1.1 Direct Digital Signature

The direct digital signature involves only the communicating parties (source, destination). It is assumed that the destination knows the public key of the source. A digital signature may be formed by encrypting the entire message with the sender's private key or by encrypting a hash code of the message with the sender's private key.

1.1.2 Arbitrated Digital Signature

The problems associated with direct digital signatures can be addressed by using an arbiter. As direct signature schemes, there are a number of arbitrated signature schemes. In general terms, they all operate as follows. Every signed message from a sender X to a receiver Y goes first to an arbiter A, who subjects the message and its signature to a number of tests to check its origin and content. The message is then dated and sent to Y with an indication that it has been verified to the satisfaction of the arbiter. The presence of A solves the problem faced by direct signature schemes: that X might disown the message.

1.2 Blind Signature:

Blind signature scheme, first introduced by David Chaum in 1983 [14, 15], allows a person to get a message signed by another party without revealing any information about the message to the other party. Using RSA signatures, Chaum demonstrated the implementation of this concept as follows: Suppose Alice has a message m that she wishes to have signed by Bob, and she does not want Bob to learn anything about m. Let (n, e) be Bob's public key and (n, d) be his private key. Alice generates a random value r such that gcd(r, n) = 1 and sends $x = (r^e m) \bmod n$ to Bob. The value x is ``blinded'' by the random value r; hence Bob can derive no useful information

from it. Bob returns the signed value $t = x^d \bmod n$ to Alice. Since $x^d_0 (r^e m)^d_0 \; r \; m^d \bmod n$. Alice can obtain the true signature s of m by computing $s = r^{-1} t \bmod n$. [14,15]

Now Alice's message has a signature she could not have obtained on her own. This signature scheme is secure provided that factoring and root extraction remains difficult. However, regardless of the status of these problems the signature scheme is unconditionally "blind" since r is random. The random r does not allow the signer to learn about the message even if the signer can solve the underlying hard problems.

1.3 What is Biometrics?

Biometrics is the science of measuring physical properties of living beings. It is the automated recognition of individuals based on their behavioral and biological characteristics like face, blood, finger-prints whichever has some unique feature. **Biometric features** are information extracted from biometric samples given above which can be used for comparison with a biometric reference. Example: characteristic measures extracted from a face photograph such as eye distance or nose size etc.

The aim of the extraction of biometric features from a biometric sample is to remove any redundant information which does not contribute to biometric recognition. This enables a fast comparison, an improved biometric performance, and may have privacy advantages. Our project uses the biometric feature matrix for the generation of the private key. [7, 13] First we can classify those schemes into key derivation (generation) and signature generation and verification framework of the matrix. The key derivation schemes imply that the signature key is derived directly from biometrics while the key authentication schemes mean that the signature key is accessed by biometric authentication.

Biometric data are directly mapped into a unique and repeatable binary string and then are transformed into a cryptographic key which is used as private key in our proposed project. No biometric template would be needed to store. But these methods are not flexible, for biometric characteristics are unique and permanent and expected to generate unique key, but in different application scenarios, a user possibly wants to use different keys [13, 11]. The hardest problem of this model is that the biometric data of a person vary dramatically depending on the acquisition method, acquisition environment, user's interaction with the acquisition device, and

(in some cases) variation in the traits due to various pathophysiological phenomena. It cannot be guaranteed to generate the same, unique key every time from different biometric samples.

The private key is generated by hashing a user's personal secret and a biometric template. If we use only the biometric template for private key generation, we always get to generate the same key since the biometric data is unique. Moreover, if the private key is disclosed, the user's biometric data cannot be used any more. Therefore, in order to cancel and regenerate the private key, the user's personal secret is also required in generation of the private key. The private key is generated by hash function such as MD5 or SHA-1 on the biometric template with the personal secret [7, 11, 13]. We have implemented the biometric feature by using the MD5 hashing technique. If we use only the biometric template for private key generation, we always get to generate same key since the biometric data is unique. Therefore, in order to cancel and regenerate the private key, the user's personal secret is also required in generation of the private key [13].

1.4 Objective:

The objective of the proposed project is to enhance the security of a system by the use of both biometric entity and cryptography and blinding. Our motive is to implement the project proposed taking the specimen of a figure print image as biometric data and generating large prime numbers under some conditions. This system not only gives security but also gives authentication to the requester party.

1.5 Motivation:

The biometric security and the blinding of the message have brought a significant change in the field of cryptography and computer security. In each and every area of the world due to importance of security there is a need of development of biometrics and blinding operation. The proposed scheme gives a mixture of both the ideas to give a better security to a system due to use of biometrics and complex mathematical techniques which are very difficult to decode.

Chapter2

Cryptographic Techniques

2.1 What is Cryptography?

Cryptography is the modern technique of converting ordinary text to unintelligible text. The ordinary text is otherwise known as plain text and unintelligible text is called cipher text. This is also known as encryption. In reverse decryption is converting the cipher text back to its original form. In the past the cryptography referred only encryption and decryption of messages by the use of a common secret key. But now-a-days due to advancement of the technology three different standard mechanisms are proposed. They are [11, 12]

- Symmetric Key cryptography
- Asymmetric Key cryptography
- Hashing

2.1.1. Symmetric Key Cryptography:

In this technique the sender encrypts the message using some encryption algorithm and some secret key which is known only to both sender and receiver parties. Then the receiver receives the message and decrypts the message using a decryption algorithm and the same secret key.

2.1.2 Asymmetric Key Cryptography:

The technique is also known as public-key-cryptography. In this case we have the same situation as that of the symmetric key cryptography with few exceptions. Firstly there are two keys involved in this- public key and private key. The sender first uses the public key for encryption and the receiver uses his private key for decrypting the message as shown in figure 2.1[11, 12]

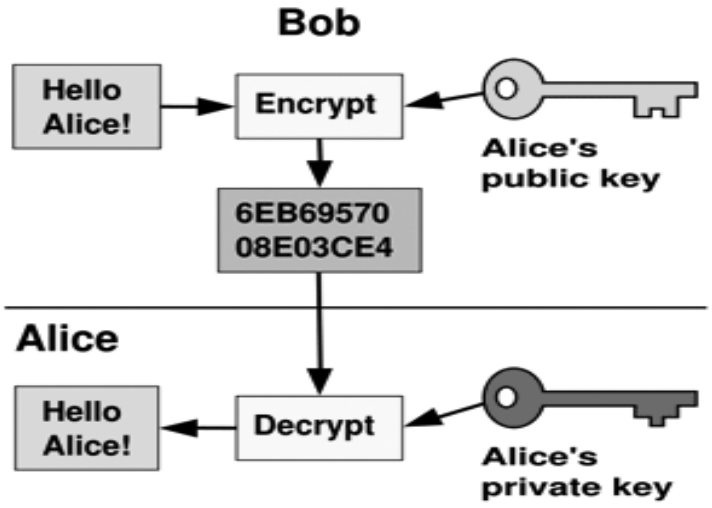

Figure 2.1-Asymmetric Key Cryptography [11, 12]

2.1.3 Hashing:

In this case a fixed length message digest created out of the variable length message. The digest is much smaller than that of the message. To be useful both the message and the digest must be sent to the receiver. Hashing is used to provide check values for the integrity of the message.

Our proposed project is uses the benefits of public-key-cryptosystem and hashing for a secured communication between two parties. A set of cryptographic hash functions uses compression function. These hash functions include RSA, MD etc. Our proposed idea uses the MD5 algorithm for message compression to convert a message to a 128 bit hexadecimal form. [11, 12]

2.2 Cryptanalysis:

Cryptanalysis is the study of methods for obtaining the meaning of encrypted information, without access to the secret information which is normally required to do so. Typically, this involves knowing how the system works and finding a secret key. In non-technical language, this is the practice of **code breaking** or **cracking the code**, although these phrases also have a specialized technical meaning. "Cryptanalysis" is also used to refer to any attempt to circumvent the security of other types of cryptographic algorithms and protocols in general, and not just encryption. However, cryptanalysis usually excludes methods of attack that do not primarily

target weaknesses in the actual cryptography, although these types of attack are an important concern and are often more effective than traditional cryptanalysis. [11, 12]

The **International Telecommunication union-Telecommunication standardization Sector (ITU-T)** provides some security services and some mechanism to implement those services.

2.3 Security Services:

The security services include:

- Data Confidentiality
- Data Integrity
- Authentication
- Non repudiation
- Access Control

2.3.1 Data Confidentiality:

Confidentiality has been defined by the International Organization for Standardization (ISO) in ISO-17799 as "ensuring that information is accessible only to those authorized to have access" and is one of the cornerstones of information security. Confidentiality is one of the design goals for many cryptosystems, made possible in practice by the techniques of modern cryptography.

It is designed to protect data from disclosure attack. The service as defined by X.800 is very broad and encompasses confidentiality of whole message or part of a message and also protection against traffic analysis. That is it is designed to prevent snooping and traffic analysis

[11, 12]

2.3.2 Data Integrity:

Data Integrity is designed for the protection of data from unauthorized modification, insertion, deletion and replaying by an advisory. It can protect the whole message or the part of message.

2.3.3 Authentication:

This service provides the authentication of the party at the other end of the line. In the connection oriented communication, it provides the authentication of the sender or receiver during the connection establishment (peer entity authentication). In connectionless communication, it authenticates the source of data (also called data origin authentication).

2.3.4 Non-repudiation:

Non-repudiation service protects against repudiation by either the sender or the receiver of the data. In this with the proof of origin, the receiver of the data can later prove the identity of the sender. If denied. In non-repudiation with the real proof of delivery the sender of the data can later prove the data were delivered to the intended recipient [11, 12].

Non-repudiation is the concept of ensuring that a party in a dispute cannot repudiate, or refute the validity of a statement or contract. Although this concept can be applied to any transmission, including television and radio, by far the most common application is in the verification and trust of signatures.

2.3.5 Access Control:

Access control is a system which enables an authority to control access to areas and resources in a given physical facility or computer-based information system. An access control system, within the field of physical security, is generally seen as the second layer in the security of a physical structure. It provides security against unauthorized access against data. The term access in this definition is very broad and can involve reading, writing, modifying, executing programs [11, 12].

2.4 Security Mechanism:

Security mechanisms include:

- Encipherment
- Data Integrity
- Authentication exchange

- Traffic padding
- Routing control
- Notarization
- Access Control
- Digital Signature

2.4.1 Encipherment:

Encipherment, hiding or covering data can provide confidentiality. It can also be used to complement other mechanisms to provide other services. Today two technique-cryptography and steganography -are used for enciphering. [11, 12]

2.4.2 Data Integrity:

The data Integrity mechanism appends to the data a short check value that has been created by a specific process from the data itself. The receiver receives the data and the check value from the received data and compares the newly created check value with the one received. If the two check values are the same the integrity of the data has been preserved. All characteristics of the data including business rules, rules for how pieces of data relate dates, definitions and lineage must be correct for data to be complete. [11, 12]

2.4.3 Authentication Exchange:

In authentication exchange, two entities exchange some messages to prove their identity to each other. For example one entity can prove that he knows a secret that only he is supposed to know.

2.4.4 Traffic Padding:

Traffic padding means inserting some bogus and unneeded data into the data traffic to thwart the advisory's attempt to use the traffic analysis. [11, 12]

2.4.5 Routing Control:

Routing control means selecting and continuously changing different available routes between the sender and the receiver to prevent the opponent from eavesdropping on a particular route. [11,12]

2.4.6 Notarization:

It means selecting a trusted third party to communicate between two entities. This can be done to prevent repudiation. The receiver can involve a third party to store the sender request in order to prevent the sender from later denying that she has made the request. [11, 12]

2.4.7 Access Control:

Access control uses methods to prove that a user has access right to the data or resources used by the system. **Access control** is a system which enables an authority to control access to areas and resources in a given physical facility or computer-based information system. An access control system, within the field of physical security, is generally seen as the second layer in the security of a physical structure. Examples of proofs are PINs and passwords. [11, 12]

2.4.8 Digital signature:

A **digital signature** or **digital signature scheme** is a mathematical scheme for demonstrating the authenticity of a digital message or document. A valid digital signature gives a recipient reason to believe that the message was created by a known sender, and that it was not altered in transit. Digital signatures are commonly used for software distribution, financial transactions, and in other cases where it is important to detect forgery and tampering.

Digital signatures are often used to implement electronic signatures, a broader term that refers to any electronic data that carries the intent of a signature, but not all electronic signatures use digital signatures. In some countries, including the United States, and members of the European Union, electronic signatures have legal significance. However, laws concerning electronic signatures do not always make clear whether they are digital cryptographic signatures in the sense used here, leaving the legal definition, and so their importance, somewhat confused.

Digital signatures employ a type of asymmetric 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. Digital signatures are equivalent to traditional handwritten signatures in many respects; properly implemented digital signatures are more difficult to forge than the handwritten type of signatures. Digital signature schemes in the sense used here are cryptographically based, and must be implemented properly to be very effective. Digital signatures can also provide non-repudiation, meaning that the signer cannot successfully claim they did not sign a message, while also claiming their private key remains secret; further, some non-repudiation schemes offer a time stamp for the digital signature, so that even if the private key is exposed, the signature is valid nonetheless. Digitally signed messages may be anything representable as a bit string: examples include electronic mail, contracts, or a message sent via some other cryptographic protocol. [5, 11, 12]

Chapter3

Modular Arithmetic and Prime Numbers

3.1 Groups, Rings, and Fields:

Groups, rings, and fields are the fundamental elements of a branch of mathematics known as abstract algebra, or modern algebra. In abstract algebra, we are concerned with sets on whose elements we can operate algebraically; that is, we can combine two elements of the set, perhaps in several ways, to obtain a third element of the set. These operations are subject to specific rules, which define the nature of the set. By convention, the notation for the two principal classes of operations on set elements is usually the same as the notation for addition and multiplication on ordinary numbers. However, it is important to note that, in abstract algebra, we are not limited to ordinary arithmetical operations. All this should become clear as we proceed.

3.1.1 GROUPS:

A **group** G sometimes denoted by {G, ·} is a set of elements with a binary operation, denoted by ·, that associates to each ordered pair (a, b) of elements in G an element (a · b) in G, such that the following axioms are obeyed: The operator · is generic and can refer to addition, multiplication, or some other mathematical operation.

(A1) Closure: If a and b belong to G, then a · b is also in G.

(A2) Associative: a · (b · c) = (a · b) · c for all a, b, c in G.

(A3) Identity element: There is an element e in G such that a · e = e · a = a for all a in G.

(A4) Inverse element: For each a in G there is an element a' in G such that a · a' = a' · a = e.

If a group has a finite number of elements, it is referred to as a **finite group**, and the **order** of the group is equal to the number of elements in the group. Otherwise, the group is an **infinite group**. A group is said to be abelian if it satisfies the following additional condition:

(A5) Commutative: a · b = b · a for all a, b in G. [11,12]

.2 RING:

ing R, sometimes denoted by {R, +, x}, is a set of elements with two binary operations, called ition and multiplication, such that for all a, b, c in R the following axioms are obeyed:

ıerally, we do not use the multiplication symbol, x, but denote multiplication by the catenation of two elements.

1-A5) R is an abelian group with respect to addition; that is, R satisfies axioms A1 through A5. r the case of an additive group, we denote the identity element as 0 and the inverse of a as a.

[1) Closure under multiplication:	If a and b belong to R, then ab is also in R.
[2) Associativity of multiplication:	a(bc) = (ab)c for all a, b, c in R.
[3) Distributive laws:	a(b + c) = ab + ac for all a, b, c in R. (a + b)c = ac + bc for all a, b, c in R.

essence, a ring is a set in which we can do addition, subtraction [a b = a + (-b)], and ltiplication without leaving the set.

ing is said to be **commutative** if it satisfies the following additional condition:[11,12]

[4) Commutativity of multiplication: ab = ba for all a, b in R.

.3 INTEGRAL DOMAIN:

ext, we define an integral domain, which is a commutative ring that obeys the following axioms:

[5) Multiplicative identity: There is an element 1 in R such that a1 = 1a = a for all a in R.

[6) No zero divisors: If a, b in R and ab = 0, then either a = 0 or b = 0. [11]

3.1.4 FIELD:

A **field** F, sometimes denoted by {F, +, x}, is a set of elements with two binary operations, called addition and multiplication, such that for all a, b, c in F the following axioms are obeyed:

(A1M6) F is an integral domain; that is, F satisfies axioms A1 through A5 and M1 through M6.

(M7) Multiplicative inverse: For each a in F, except 0, there is an element a^{-1} in F such that $aa^{-1} = (a^{-1}) a = 1$. [11,12]

3.2 GALOIS FIELD:

3.2.1 Properties Of Congruence:

1. $a \equiv b \pmod{n}$ if $n|(a\ b)$.
2. $a \equiv b \pmod{n}$ implies $b \equiv a \pmod{n}$..
3. $a \equiv b \pmod{n}$ and $b \equiv c \pmod{n}$ imply $a \equiv c \pmod{n}$.[11]

3.2.2 Modular Arithmetic Operations:

Modular arithmetic exhibits the following properties:

1. $[(a \bmod n) + (b \bmod n)] \bmod n = (a + b) \bmod n$
2. $[(a \bmod n)\ (b \bmod n)] \bmod n = (a\ b) \bmod n$
3. $[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$

We defined a field as a set that obeys all of the axioms of given above and gave some examples of infinite fields. Infinite fields are not of particular interest in the context of cryptography. However, finite fields play a crucial role in many cryptographic algorithms. It can be shown that the order of a finite field (number of elements in the field) must be a power of a prime p^n, where n is a positive integer.. Here, we need only say that a prime number is an integer whose only positive integer factors are itself and 1. That is, the only positive integers that are divisors of p are p and 1. [11]

The finite field of order p^n is generally written **GF (p^n)**; stands for **Galois field**, in honor of the mathematician who first studied finite fields. Two special cases are of interest for our purposes. For n = 1, we have the finite field GF (p); this finite field has a different structure than that for finite fields with n > 1, we look at finite fields of the form GF (2^n).

We mentioned that the order of a finite field must be of the form p^n where p is a prime and n is a positive integer; we looked at the special case of finite fields with order p. We found that, using modular arithmetic in Z_p, all of the axioms for a field are satisfied. For polynomials over p^n, with n > 1, operations modulo p^n do not produce a field. In this section, we show what structure satisfies the axioms for a field in a set with p^n elements, and concentrate on GF (2^n). [11]

Virtually all encryption algorithms, both symmetric and public key, involve arithmetic operations on integers. If one of the operations that is used in the algorithm is division, then we need to work in arithmetic defined over a field. For convenience and for implementation efficiency, we would also like to work with integers that fit exactly into a given number of bits, with no wasted bit patterns. That is, we wish to work with integers in the range 0 through 2^n 1, which fit into an n-bit word. Intuitively, it would seem that an algorithm that maps the integers unevenly onto themselves might be cryptographically weaker than one that provides a uniform mapping. Thus, the finite fields of the form GF (2^n) are attractive for cryptographic algorithms.

To summarize, we are looking for a set consisting of 2^n elements, together with a definition of addition and multiplication over the set that define a field. We can assign a unique integer in the range 0 through 2^n -1 to each element of the set. Keep in mind that we will not use modular arithmetic, as we have seen that this does not result in a field.[11,12]

3.3 PRIME NUMBER GENERATION AND TESTING:

Asymmetric key cryptography uses primes extensively. The positive numbers can be classified into three groups: the number 1, primes, and composites. A positive integer is prime if and only if it is exactly divisible by two integers, 1 and itself. The use of public-key cryptography is pervasive in the information protection and privacy arenas. Public key crypto algorithms utilize prime numbers extensively; indeed, prime numbers are an essential part of the major public key systems.

3.3.1 Relatively Prime Numbers:

Relatively Prime Numbers: two integers are called relatively prime to one another if they have no common factors other than 1. The numbers themselves need not be prime. In formal notation this is expressed as

gcd(M, N) = 1

i.e., the greatest common devisor (largest common factor) of the two numbers is 1. For example, 8 and 15 are relatively prime because they have no common factors other than 1. 12 and 15 are not relatively prime because they share the common factor 3. Relatively prime numbers are important in asymmetric cryptography; it is important to understand the difference between prime numbers and relatively prime numbers and understand that two numbers that are not prime (e.g., 8 and 15) may still be relatively prime. All prime numbers are by definition relatively prime to one another. [10, 12]

3.3.2 Mersenne Primes

Any positive integer that is one less than an integral power of 2 can be expressed as $2^n - 1$,
Where n is a positive integer. Many such numbers are prime. Put another way, it turns out that some integers that are exactly one less than an even power of 2 are prime. The concept of expressing prime numbers in the form $2^n–1$ dates to antiquity, but the first major work published about such primes was authored by a French monk named Marin Mersenne (1588-1648) in the 17th century and prime numbers that can be expressed in this form are now called Mersenne Primes. For simplicity we will henceforth use Mn to represent the Mersenne prime $2^n –1$. For example, M3 = 2^3-1 = 7, the second Mersenne prime. The search for ever larger Mersenne primes is an icon of today's recreational mathematical pursuits. In late 2001 the largest Mersenne prime yet discovered was verified by a 20-year-old resident of Ontario, Canada; he was participating in a distributed search managed by software provided by Entropia, Inc. The number has 4,053,946 digits – yes, that's the number of digits in the Mersenne prime expressed in

decimal, not the number itself. The Mersenne prime is $2^{13,466,917}$– 1 an incomprehensibly huge number. Early writings about Mersenne primes conjectured that 2^n-1 is prime for all primes n; i.e., it was believed that one less than 2n is prime for every case where n is prime. In 1536, however, it was shown that M11 is composite.

M11 = 2^{11}-1 = 2047 = 23 * 89

A number of later writings contained incorrect assertions about which Mn values are prime, including those of Mersenne himself. Mersenne asserted in 1644 that Mn is prime for

n = 2, 3, 5, 7, 13, 17, 19, 67, 127, 257

And that all other Mn is composite for n < 257. This is incorrect (the values above 19 are actually 31, 61, 89, 107 and 127; the next is 521), though the errors were not discovered until hundreds of years later (Euler added 31 to the list in 1750). There are many interesting theorems related to Mersenne numbers and primes. For example, if Mn is prime then n is also prime (though, as noted above, the converse is not necessarily true). These theorems are very useful in Modern number theory and factoring investigations. Such investigations might one day yield an efficient algorithm for factoring very large numbers into their prime factors, a breakthrough (or not, depending upon how one looks at it) that would render most contemporary public key cryptography systems useless.[10,12]

3.3.3 Fermat Little Theorem

The first probabilistic, method we discussed in the Fermat Primality test:

$$\textit{If n is a prime, then } a^{n-1} \equiv 1 \pmod{n}$$

Note that this means if n is prime, the congruence holds. It does not mean that if the congruence holds, n is prime. The integer can be prime or composite. We can define the following as Fermat's test:[11,12]

$$\textit{If n is a prime, then } a^{n-1} \equiv 1 \bmod n$$

$$\textit{If n is composite, it is possible that } a^{n-1} \equiv 1 \bmod n$$

All primes pass the Fermat's test. Composite may also pass the Fermat's test as well. The bit operation complexity of Fermat's test is same as the complexity of an algorithm that calculates the exponentiation.

3.3.4 Square Root Test:

In modular arithmetic, if n is a prime the square root of 1 is either +1 or -1. If n is composite the square root is +1 or -1, but there may be other roots. This is known as square root Primality test[11,12].

If n is a prime, sqrt (1) mod n=+1 or-1

If n is a composite, sqrt (1) mod n=+1 or-1 and possibly other values.

3.3.5 Miller-Rabin Primality Test:

The Miller-Rabin Primality test combines the **Fermat's test** and **square-root test** in a very elegant and efficient way to find a strong pseudo prime (a prime with a very high probability of being a prime). In this test we write n-1 as the product of an odd number and a power of two.

$$n-1 = m * 2^k$$

In other words, instead of calculating a^{n-1} (mod n) in one step, we can do it in k+1 steps. The benefit is that in each step, the square root test can be performed. If the square root test fails we stop and declare that n is a composite number. In each step we assure ourselves that the Fermat's test is passed and the square root test is satisfied between all pairs of adjacent steps, if applicable. It is a probabilistic method. There exists a proof that each time the number passes the Miller-Rabin Primality Test, the probability that it is not a prime is ¼. If the number passes m tests (with m different bases) the probability that it is not a prime is $(1/4)^m$. [10, 12]

Algorithm of Miller-Rabin Primality Test:

1. ***Miller-Rabin_Test(n, a)***
2. ***{***
3. ***Find m and k such that*** $n-1=m * 2^k$
4. $T \leftarrow a^m \bmod n$
5. ***If (T=1 || T-n=-1)***
6. ***return "a prime"***
7. ***for(i=1 to k-1)***
8. ***{***
9. $T \leftarrow T^2 \bmod n$
10. ***if (T=1) return "a composite number"***
11. ***if (T-n=-1) return "a prime number"***
12. ***}***
13. ***return "a composite number"***
14. ***}*** [11,12]

3.3.6 Pollard p-1 Factorization Method:

In 1974, John M. Pollard developed a method that finds a factor p of a number based on the condition that p-1 has no factor larger than a predefined value B, called the bound. Pollard showed that in this case

$$p=gcd\ (2^{B!}-1,\ n)$$

The algorithm shows the pseudo code for Pollard p-1 factorization method. He we should note that when we come out of the loop, $2^{B!}$ is stored in a.[11,12]

ALGORITHM

1. ***Pollard_p-1_factorisation (n ,B)***
2. ***{***
3. ***a←2***
4. ***e←2***
5. ***while (e <=B)***
6. ***{***
7. ***$a \leftarrow a^e \bmod n$***
8. ***e←e+1***
9. ***}***
10. ***p=gcd(a-1,n)***
11. ***If (1<p<n)***
12. ***return p***
13. ***return failure***
14. ***}*** [11,12]

3.4 Chinese Remainder Theorem:

The Chinese Remainder Theorem (CRT) is used to solve a set of congruence equations with one variable and different moduli, which are relatively prime.

$$x \equiv a_1 \pmod{m_1}$$

$$x \equiv a_2 \pmod{m_2}$$

$$\ldots.$$

$$x \equiv a_k \pmod{m_k}$$

The Chinese Remainder Theorem states that the above equations have a unique solution if the moduli are relatively prime.[12]

Example:

$x \equiv 2 (\text{mod } 3)$

$x \equiv 3 (\text{mod } 5)$

$x \equiv 5 (\text{mod } 7)$

the solution to this set of equations is x=23

Solution to the problem:

The solution to the set of equation follows the following steps:

1. Find $M = m_1 * m_2 * m_3 * \ldots * m_k$. This is the common modulus.
2. Find $M_1 = M / m_1$, $M_2 = M / m_2$, $\ldots$, $Mk = M / m_k$.
3. Find the multiplicative inverse of $M_1, M_2, \ldots, M_k$ using the corresponding moduli ($m_1, m_2, \ldots, m_k$) . Call the inverses $M_1^{-1}, M_2^{-1}, \ldots, M_k^{-1}$.
4. The solution to the simultaneous equation is

$$x = (a_1 * M_1 * M_1^{-1} + a_2 * M_2 * M_2^{-1} + \ldots + a_k * M_k * M_k^{-1}) \bmod M$$

Note that the set of equations can have a solution even if the moduli ($m_1, m_2, \ldots, m_k$) are not relatively prime but meet other conditions. The Chinese Remainder Theorem has several applications in cryptography. One is to solve quadratic congruence and the other is to represent a very large number in terms of a list of small integers.[12]

Chapter4

Implementation and results

4.1 INITIALIZATION :

The Initialization phase starts with initialization of parameters and ends with generation of all Private and Public keys.

The initialization phase involves following sub phases

- Generation of Prime numbers
- Galois Field Concept to find g
- Biometric Feature Extraction
- Normalization and use of hash function for calculation of Private key d1
- Determination of Public key Y

We go through the sub phases as follows

4.1.1 Generation of Prime numbers

Use of prime numbers is one of the most important issues for security purpose in cryptographic work. Prime numbers works like base for the generation of private keys and public keys in cryptographic system. This Prime number generation sub phase consists of Prime number generation and Primality test stages.

For generating prime numbers we first generate prime numbers using the common algorithm for generating first 100 prime numbers and store these prime numbers in an array to use them in generating large prime numbers using following generation method

Mersenne Prime generation method:

The above method involves the relation

$\mathbf{Mp = 2^p - 1}$ where Mp– Large prime number generated using the

Prime number p (stored prime number initially)

We iteratively use the above method of prime number generation to generate large prime numbers. As the Mersenne Prime number generation method is a probabilistic method, it is revealed that the prime numbers generated in this method are not all primes.[10,12]

For example:

We found the prime number 2087 by using this method. But 2087 is not a prime number. Hence for checking the numbers generated by Mersenne Prime number generation method for being prime numbers or not we go for Miller-Rabin Primality test.

Miller-Rabin Primality Test:

As described in 3.3.5 it is also a probabilistic method which checks a prime number is actually prime or not. So we implement Miller-Rabin Primality test method and checks all the prime numbers generated before to check their Primality.

The Prime number after going through Primality test is stored .The value of p and q are chosen from the prime numbers stored for the calculation of N. We iteratively use permutation and combination method to find the prime numbers p and q to calculate N satisfying the following condition [11]

N = 4pq + 1 where N= Large Prime number

We got two Prime numbers p and q as 11 and 23. Then we got N=4×11×23+1=1013 (which is also a prime number)

We use BigInteger class which is present in **Java.math.BigInteger** package for storing large Prime numbers. After this we go for determining the value of g which is an element over the GF (N).

4.1.2 Galois Field Concept to find g:

As described in Chapter 3.2 the finite field of order p^n is generally written GF (p^n); stands for Galois field, in honor of the mathematician who first studied finite fields. Two special cases are of interest for our purposes. For $n = 1$, we have the finite field GF (p); this finite field has a

different structure than that for finite fields with n > 1, we look at finite fields of the form GF (2n)

For our project purpose we define the finite Galois field GF (N) and we choose g such that g belongs to GF(N) and under the conditions that

$$\mathbf{g^{pq} \equiv 1 \pmod N}$$

$$\mathbf{g^{p} \neq 1 \pmod N}$$

Going through the above conditions we found g=16 as [4, 6]

$$16^{23\times 11}\, mod\ 1023 = 1\ and$$

$$16^{11}\, mod\ 1023 \neq 1$$

4.1.3 Biometric Feature extraction:

As described in Chapter1 the biometric information's like finger prints, iris, retina DNA, tissue and other features whatever its kind which are unique to an individual are embedded into private key d1.

We use the finger print as our biometric information and the finger print of signer is taken from which the biometric features are extracted. We take the finger print of signer and extract the RGB values by scanning through the image associated with each pixel. Each Pixel in the image of the finger print consists of 3 field values for RGB which is determined by the intensity of that pixel.

Figure 4.1 Finger print

Here the intensity values associated for different pixel in finger print figure 4.1 is unique for different persons which make our system more and more secure. For generating the RGB values from the finger print image, for implementation purpose we use javax.imageIO.* package. From the package we use following two classes

ImageIO class

ImageReader class

We define **getRGB ( )** method to calculate the RGB values associated for each pixel.

4.1.4 Normalization and use hash function for calculation of Private key d1:

After getting the RGB values our primary need is to normalize those RGB values. By this method of normalization we take a mid level RGB value above which we consider as 1 and below as 0.Thus we get a unique binary value associated with the finger print of the signer. The

binary value we got is considered as the input to the hash function for determining the private key d1.We use the MD5 -128 hash function upon the binary value. [11, 12]

MD5-128 hash function is based on the algorithm that

- The message to be hashed is divided into blocks of 512 bits.
- If necessary go for padding in the message to form the total bits of message to the multiple of 512.
- Choose a 128-bit initial key.
- A message block (512 bits) is compressed along with the initial key to form the 128-bit next key which is used as the key for the next block.
- The next key produced is called the message digest of the message block which is compressed.
- The message digest for the last block is considered as the hash value of the message.

In this way we get a 128-bit hexadecimal value. [11, 12]

In MD5-128 hash function code we use **MessageDigest class which** is present in **java.security package.**

In the above class there are 2 methods which are used to generate the hash values. The two methods are

getInstance( )

digest ( )

After applying the MD5 hashing function upon our normalized binary biometric feature value we got the hexadecimal value:

o/p **: 9d7183f16acce70658f686ae7f1a4d20**

Then we go for converting the hexadecimal value into decimal value which is the value of d1 (one of the private key).The value of d1 we found from the above finger print as

d1= **209278201005944594766464979853618335008**

After getting the value of d1 we go for determination of the other private key d2 as follows.

We choose the value of d2 under the constraint 1 < d2 < ***pq***/2 i.e 1< d2 <253/2 (where ***pq***=253)

We choose d2= 89 [6, 9]

After calculating Private keys d1,d2 and the values of g and N ,we go for calculating the Public key Y value.

4.1.5 Determination of Public key Y:

The public key is generated applying the following formula

$$\mathbf{Y= g^{(d1+d2)\hat{}2+(d1+d2)\hat{}-2}} \bmod N$$

=16

After generating the Public Key Y, the signer sends public key(Y), Hash function H (.) and N to the requester as public data. Then the message blinding phase by the requester is done.[6, 9]

4.2 BLINDING PHASE:

This phase consists of three sub phases which are carried out by both signer and requester

- Generation of r and k by Signer
- Message blinding by requester
- Generation of hashed message

The sub phases are described as follows

4.2.1 Generation of r and k by Signer:

This sub phase is completely done by signer. The signer choose a random value k such that 1< k < ***pq***/2.We take the value of k here as 45.After computing k the signer goes for calculating r applying the following formula.

$$r = g^{(k^2+(1/k)^2)} \bmod N$$

The r value is found to be 16 [6, 7, 11, 13]

After calculating the value of r the signer sends it to the requester for the blinding of the requester message.

4.2.2 Message blinding by requester:

The requester gets the public key(y), N, H (.), and r value from the signer. Using the values the requester goes for message blinding. In message blinding the requester choose a message m and blinds it using the following formula

$$\mathbf{M = m^b \bmod N}$$

Where M is the message blinded

b is the randomized factor belongs to Zq+ [11]

The value of b is chosen as 45

We get the blinded message M = 308 [14]

4.2.3 Generation of hashed message:

This sub phase is done by the requester completely. After blinding the message i.e after getting M the requester concatenates the M value and r. Then the same hash function sent by signer (MD5-128) is applied over the concatenated M and r value to get H(M, ,r). After applying the MD5-128 hash function we got a 128 bit hexadecimal value. We convert the hexadecimal value into decimal value which is our H(M, ,r) value.

The H(M, ,r) value is calculated by the requester as 30816.After calculating the H(M, ,r) value the requester sends it to the signer for generation of digital signature S.[11,12]

4.3 SIGNING

Signing phase refers to the generation of digital signature and sending this signature to the requester for signing purpose. This phase is completely done by the signer. The signer has the private keys d1, d2, the values N, k, r with her and gets the blinded message H(M, ,r) from the requester. Using the above values signer computes the Digital signature S applying the following modular quadratic equation.

$$\mathbf{H\,(M, ,r) = (\,(d1+d2)+\,(d1+d2)^{-1})^{\,2}\,r + (k+k^{-1})^{\,2}\,S^2 \bmod pq}$$

The above modular quadratic equation is solved by using quadratic residue and quadratic Congruence concept.[6,9]

4.3.1 Quadratic Residue

To solve the above equation we need the help of quadratic residue. There is a equation the type $\mathbf{x^2 \equiv a \pmod p}$

Such that (a,p)=1 ,then *a* is called the quadratic residue of *p* if the above equation is solvable, otherwise *a* is a quadratic nonresidue.

Euler's Criterion:

Let p be an odd prime. Then a positive integer a with p┼a, is a quadratic residue of p if and only if

$$\mathbf{a^{(p-1)/2} \equiv 1 (mod\ p)}$$

We check for the above criterion for our equation

$$\boldsymbol{H(M, ,r) = ((d_1 + d_2) + (d_1 + d_2)^{-1})^2\, r + (k + k^{-1})^2\, S^2 \bmod pq}$$

We have $\boldsymbol{a=(H(M, ,r)- ((d_1 + d_2) + (d_1 + d_2)^{-1})^2\, r)/ (k + k^{-1})^2}$*……....(eq 1)*

$\boldsymbol{p= pq}$*. .. . . (eq 2)*

So we check for the Euler's criterion for the above a and p. After it is satisfied we go for the next step which is called solvable test.

Solvability Check:

Legendre Symbol

If p is an odd prime and a be any integer such that p┼a, then the Legendre symbol (*a/p*) is defined as

(a/p)={1 if a is a quadratic residue of p -1 if a is quadratic nonresidue of p}

For example:

13 has quadratic residues 1, 3, 4,9,10 and 12 and nonresidues 2,5,6,7,8,11 [16]

Hence (1/13)= (3/13)= (4/13)= (9/13)= (10/13)= (12/13)=1 whereas (2/13)= (5/13)= (6/13)= (7/13)= (8/13)= (11/13)=-1

So for Solvability check for the equation $x^2 \equiv a \pmod{pq}$ we have to check whether (a/p)=1=(a/q).

In our project equation the value of pq is 253.

253=11×23

Hence p=11, q=23

So we check (a/11)=1=(a/23) (a is defined in eq1)

We use another corollary for getting the solutions……(i)

If $p \equiv 3 \pmod 4$ and $x^2 \equiv a \pmod p$ is solvable then we can explicitly use the formula

$$x \equiv \pm a^{(p+1)/4} \pmod p$$

So we apply the above corollary to get the solutions for the equations

$S^2 \equiv a \pmod p$ and $S^2 \equiv a \pmod q$ [16]

We got the solution as

$S \equiv \pm a^{(p+1)/4} \pmod{p}$ and

$S \equiv \pm a^{(p+1)/4} \pmod{q}$

Then we go for Chinese remainder Theorem (CRT) to get the solution for 'S' for mod pq.

4.3.2 Application of Chinese Remainder Theorem (CRT):

Chinese Remainder Theorem is used for solving linear system of Congruences i.e $x \equiv a_i \pmod{m_i}$,where moduli are pair wise relatively prime and $1 \leq i \leq k$, has a unique solution modulo $m_1\, m_2 \ldots m_k$.[11, 12]

From Chinese Remainder Theorem we can get the solution

$$x \equiv \sum a_i M_i y_i \pmod{M}$$

where a_i=remainder of ith modular equation

$$M = m_1 + m_2 + \ldots + m_i$$

$$M_i = M/m_i$$

and $M_i y_i \equiv 1 \pmod{m_i}$

Here in our equation

$$\mathbf{S^2 \equiv (H(M, ,r) - ((d_1 + d_2) + (d_1 + d_2)^{-1})^2 r) / (k + k^{-1})^2 \pmod{pq}}$$

We first check $((H(M, ,r) - ((d_1 + d_2) + (d_1 + d_2)^{-1})^2 r)/(k + k^{-1})^2)/p) = 1 = ((H(M, ,r) - ((d_1 + d_2) + (d_1 + d_2)^{-1})^2 r)/(k + k^{-1})^2)/q)$

After checking this we found it to be satisfied. So the equation $S^2 \equiv (H(M, ,r) - ((d_1 + d_2) + (d_1 + d_2)^{-1})^2 r)/(k + k^{-1})^2 \pmod{pq}$ is solvable .Then we check that $p \equiv 3 \pmod 4$ and $q \equiv 3 \pmod 4$ (where p=11,q=23) .So we apply corollary (i) to get $S \equiv \pm 7\ (mod\ 11)$ and $S \equiv \pm 4\ (mod\ 23)$

After solving for S value under the mod 11 and mod 23 we go for CRT and get the solution for S under mod pq i.e 253.Then we solve for linear modular congruence for S and get the signature S.[4, 7, 12]

Hence the signer determines the digital signature and sends it to the requester for verification and further use.

4.4 UNBLINDING AND VERIFICATION

This phase is opposite to the Blinding phase. This phase is completely done by the requester. The requester gets the digital Signature S from the signer. Then it goes for verification of the signature whether it is genuinely generated by the concerned signer or not. The requester verifies the Signature by applying the following equation

$$g^{H(M,r)} \equiv Y^{r}\ r^{s^2}\ g^{2(r+s^2)} \pmod{N}$$

If the above verification equation is satisfied then the Signature is considered to be a valid signature and the requester gets the signature along with the message, otherwise the signature is rejected.[4, 6, 7]

Chapter5

Security Analysis

5.1 SECURITY ANALYSIS

One of the major tasks in a Cryptographic work is to analyze the security issues involved in that work. In our scheme we use Private and public key cryptosystem where the security about the private keys is the main concern. Here the signer keeps the private keys with her and sends the public keys and values to the requester. The main concern about the security is that if a hacker hacks the public keys and values, she should not get the private keys from those.

Our scheme is secure due to the following three criteria

1. Use of Biometric value
2. Use of Integer Factorization
3. Discrete Logarithms

5.1.1 Use of Biometric Value

Since this signature scheme is based on biometric feature values, the attacker cannot forge the biological information which is purely personal. Hence this scheme is secure and can be applicable to e-voting, e-cash and digital rights management.

5.1.2 Integer Factorization

In our scheme we take large prime number N where $N=4\times p\times q+1$.As the value N is public, to get values of p and q we need modular factorization of N. The complexity of factorization in modular arithmetic is large. It is also the fact that, no such perfect algorithm has been found yet.

The complexity of most of the factorization algorithm is exponential or in best case it is sub exponential. Hence to go for a algorithm whose complexity is of exponential order is almost impossible. As to get the private values one need to use modular factorization, our scheme is secure. [11, 12]

5.1.3 Discrete Logarithm

We get the public key Y as

$$\mathbf{Y= } g^{(\mathbf{d1+d2})^{\mathbf{2}}+(\mathbf{d1+d2})^{-\mathbf{2}}} \mathbf{\ mod\ N}$$

By getting the public value Y, hacker can go for calculating g value which is vital private value used for computation of different values. To get the value of g one should go for discrete logarithm i.e $g = log_{(d1+d2)^2+(d1+d2)^{-2}} Y$. There is no efficient algorithm for modular discrete logarithm. The complexity of available algorithm is very high which is impossible to be worked out. Hence our Scheme is secure in terms of discrete logarithm. [4, 9, 11, 12]

Chapter6

Conclusion

5.2 Conclusion and Future Work

This proposed scheme can be implemented using various biometric entities like face, retina and iris, etc. The scheme is secure as the intruder has to solve both discrete logarithm and integer factorization problems simultaneously. The signature cannot be forged as biometrics information is associated with the signature, which is unique to a signer. For these security features, the proposed scheme can also be applied in practical applications such as smart cards, anonymous electronic cash and e-voting systems. The scheme can also be applicable to Elliptical Curve Cryptography (ECC). The proposed scheme can further be implemented for partial blind signature and fair blind signature schemes.

Bibliography

[1] Y.-J. Chang, W. Zhang, and T. Chen. Biometrics-Based Cryptographic Key Generation. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, volume 3, pages 2203–2206, 2004.

[2] Y. Dodis, L. Reyzin, and A. Smith. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. In *Proceedings of Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 523–540. Springer-Verlag, 2004.

[3] F. Hao, R. Anderson, and J. Daugman. Combining Crypto with Biometrics Effectively. *IEEE Transactions on Computers*, 55(9):1081–1088, September 2006.

[4] Shao Z Signature schemes based on factoring and discrete logarithms, IEEE Proc., Comput. Digit. Tech., Vol.145, No.1, page 33-36, , 1998

[5] Camenisch, J. L. Pirereau, J.M. and Stadler, M.A. Blind Signature Based on Discrete Logarithm Problem, Advanced in Cryptology, EUROCRYPT '94**,** pp 428-432, *(1995)*

[6]] Pointcheval, D. and Stern, J. **"**New Blind Signatures equivalent to Factoring". Proceeding of 4th ACM Conf on Computer and Communication Security, ***(***1997), pp 92-99.

[7] Burnett A, Duffy A, Dowling T, "A biometric identity based signature scheme".

[8] X Liu, Q Miao, D Li, "A new Special Biometric Identity based Signature Scheme", International Journal of Security and its Applications, Vol. 2, No. 1, January 2008

[9] ELGAMAL, T.: **'**A public key cryptosystem and a signature scheme based on discrete logarithms', IEEE Trans, TT--31, nn. 469412, 1985.

[10] Great Internet Mersenne Prime Search (GIMPS) (www.mersenne.org) [14: "Researchers Discover Largest Multi-Million-Digit Prime Using Entropia Distributed Computing Grid]

[11] Stallings, W., *Cryptography and Network Security: Principles and Practice*, Fourth Edition, New Jersey: Prentice-Hall, 2005

[12] Behrouz A.Forouzan, *Cryptography and Network Security,* Special Indian Edition 2007,Tata McGraw-Hill,2007.

[13]Yunsu Chung, Kiyoung Moon, Hyung-Woo Lee: Biometric Certificate Based Biometric Digital Key Generation with Protection Mechanism. FBIT 2007: 709-714

[14] Zichen Li, Junmei Zhang, Weidong Kou, The Unlinkability of Randomization-Enhanced Chaum's Blind Signature Scheme, 0-7695-1926-1/03, 2003 IEEE

[15] D Chaum.“ Blind signatures for untraceable payments”, Advances in Cryptology Proceedings of Crypto 82, pp 199-203, 1983

[16]Thomas Koshy, *Elementary Number Theory with Application*, Academic Press , 2/e 2nd edition 2007.