An Improved Certificateless Generalized Signcryption Scheme

Jitendra Kumar Rout
(Roll No:211CS2287)

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela – 769 008, India

An Improved Certificateless Generalized Signcryption Scheme

Dissertation submitted in

June 2013

to the department of

Computer Science and Engineering

of

National Institute of Technology Rourkela

in partial fulfillment of the requirements

for the degree of

Master Of Technologyy

by

Jitendra Kumar Rout

(Roll No- 211CS2287)

under the supervision of

Prof. Banshidhar Majhi

Department of Computer Science and Engineering

National Institute of Technology Rourkela

Rourkela – 769 008, India

Dedicated to my family

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008,Odisha,India. www.nitrkl.ac.in

Certificate

This is to certify that the thesis entitled ***An Improved Certificateless Generalized Signcryption Scheme*** by *Jitendra Kumar Rout*, bearing roll number 211CS2287, is a record of an original research work carried out by him under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of *Master Of Technology* in *Computer Science and Engineering*. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Place: NIT, Rourkela
Date: 03 - 06 - 2013

Banshidhar Majhi
Professor
Department of CSE
National Institute of Technology
Rourkela-769008

Acknowledgment

The beginning of knowledge is the discovery of something we do not understand.

Frank Herbert

First of all, I would like to express my deep sense of respect and gratitude towards my advisor and guide Prof (Dr.) Banshidhar Majhi, who has been the guiding force behind this work. I want to thank him for introducing me to the field of Advanced Cryptography and giving me opportunity to work under him. Without invaluable advice and assistance, it would not have been possible for me to complete this thesis. I am greatly indebted to him for his encouragement and invaluable advice.

Secondly, I would like to thank Prof. S.K. Jena, Prof. A.K. Turuk, Prof. B D Sahoo, Prof. D P Mohapatra, Prof. S K Rath and Prof. S Chinara for their valuable suggestions, and encouragements during the research work.

I must acknowledge the academic resources that I have got from NIT Rourkela. I would like to thank administrative and technical staff members of the Department who have been kind enough to advise and help in their respective roles.

During my studies at NIT Rourkela, I made many friends. I would like to thank them all, for all the great moments I had with them.

My family is the backbone behind all my endeavors with their love and support. No word of thanks can be enough for them for their encouragement, support and belief in me.

Finally, I thank God for everything.

Jitendra Kumar Rout

Abstract

Signcryption is basically a cryptographic primitive which provides both signature and encryption functions simultaneously, but it is not useful when only one of the function is required. Generalized Signcryption (GSC) is a special cryptographic primitive which can provide Signcryption function when security and authenticity are needed simultaneously, and can also provide encryption or signature function separately when any one of them is needed. The first Generalized Signcryption was proposed in 2006 by Han et al. Since then many Generalized Signcryption has been proposed based on ECDLP, based on Bilinear Pairing, Identity based and some are also proposed in Certificateless environment. Majority of the Generalized Signcryption schemes uses Random Oracle Model for their security proof and few are proposed based on Standard model.

In this thesis we have surveyed the existing GSC schemes and compare their security properties and efficiency. Along with this we also have proposed two schemes of which first one is an Identity based Generalized Signcryption Scheme and second one is a Certificateless Generalized Signcryption Scheme which is a variation of Certificateless Signcryption Scheme by Barbosa et al. We begin by giving formal definition of GSC primitive and complete with comparative study with other models. Finally, we look ahead at what future progress might be made in the field.

Contents

List of Figures

List of Tables

Acronyms

Acronyms	Description
AES	Advanced Encryption Standard
CA	Certification Authority
BDHP	Bilinear Diffie-Hellman Problem
BPGSC	Bilinear Pairing based Generalized Signcryption
CDHP	Computational Diffie-Hellman Problem
CLGSC	Certificateless Generalized Signcryption
CRHF	Collision Resistance Hash Function
DES	Data Encryption Standard
DLP	Discrete Logarithm Problem
DSA	Digital Signature Standard
ECC	Elliptic Curve Cryptosystem
ECDLP	Elliptic Curve Discrete Logarithm Problem
ECGSC	Elliptic Curve based Generalized Signcryption
GF	Galois Field
GSC	Generalized Signcryption
HMAC	Hashed Message Authentication Code
IBGSC	Identity based Generalized Signcryption
IDGSC	ID based Generalized Signcryption
KGC	Key Generation Center
MAC	Message Authentication Code
MD	Message Digest
OWF	One-Way Function
PKG	Private key Generator
PKI	Public-Key Infrastructure
RSA	Rivest, Shamir, Adelman
SHA	Secure Hash Algorithm

Symbols and Notations

Notations	Description
$\mid$	Divides
$\parallel$	Concatenation
$=$	Equality
$\equiv$	Congruence
$\approx$	Approximately
$\forall$	For all
$\geq$	Greater than equal to
$\leq$	Less than equal to
$\neq$	Not equal to
$\oplus$	XOR operation
$\perp$	False
$\top$	True
$\times$	Multiplication
$hash()$	One way hash function
$\|p\|$	Number of bits in p
G	Group
E	Elliptic curve
mod	modulo operator
$GF(p)$	The finite field of order p
Z_p	set of non-negative integers less than p
$D_k()$	Symmetric key decryption
$E_k()$	Symmetric key encryption

Chapter 1

Introduction

In the era of information we are living, information about every aspect of life has to be kept. Information is like an asset, which has a value like other asset. As an asset it has to be secured from threats and attacks. To keep secure, information needs to be hidden from unauthorized access (confidentiality), protected from unauthorized change (Integrity), and available to authorized entity when needed (Availability) [1]. With the growth of computer networks and Internet, information now a day becomes distributed. So not only information needs to be confidential when it is stored in computer, its confidentiality should also be maintained when it is being transmitted from one computer to another.

Two of the most important functions of modern cryptography are confidentiality and data integrity. Confidentiality can be achieved by encryption techniques, whereas integrity can be provided by the use of authentication techniques. Encryption technique falls into two broad categories: private key encryption and public key encryption [2]. Similarly, authentication techniques can be categorized by private key authentication and public key digital signatures. In private (symmetric) key cryptography, a secret has to be shared between participants before any communication, which is infeasible for a large community. In asymmetric (public) key cryptography, the secret is personal (unshared); each party creates and keeps its own secret. The public key cryptography is best suited for some applications like: authentication and digital signature. Whenever an

application is based on personal secret, public key cryptography needs to be used. However, public key encryption is slower than symmetric key encryption. In public key cryptography any message that are encrypted using public key can only be decrypted by applying the same algorithm, but using the matching private key. Similarly any message that is signed by a private key can only be verified by matching public key. To check authentication of the message (proof of origin) the sender has to sign the message before it gets delivered to the recipient. Message confidentiality and senders authentication in the open channel is a basic and important need of Internet technology. Until before decade message encryption and digital signature have been viewed as important but distinct building blocks of various cryptographic systems. In public key schemes the traditional method is to digitally sign the message then encrypt it and send it to the recipient. The recipient will decrypt the message and check the authenticity of the message. This two-step sequential approach is called "Signature-then-Encryption". Disadvantage of this approach is that any arbitrary composition cannot guarantee security. This approach also has low efficiency and cost is sum of authentication and encryption. Signcryption [3] provides the solution to this problem by combining both the functionalities into a single logical step. A Signcryption scheme simultaneously fulfills the security attributes of an encryption and those of a digital signature. Though signcryption is efficient to provide both signature and encryption functions simultaneously, it will not be useful in scenarios where sometimes we need only one function separately and sometimes both the function jointly. One solution [4] to this is to combine signcryption with other signature and encryption module. That means applications must contain at least three cryptographic primitives (Signcryption, Signature and Encryption), which will be infeasible in some resource-constrained environments like: embedded systems, sensor networks, and ubiquitous computing. Solution to the problem is Generalized Signcryption [5] ,which is a cryptographic primitive that can work as an encryption scheme or a signature scheme as per the need. In other words without any additional modification and computation, it provides double functions when the confidentiality and the authenticity are required simultaneously or separately.

1.1 Message Encryption:

Encryption is the process of converting a message from comprehensive (readable) form into an incomprehensive form and back again at the receiver end, rendering it unreadable by interceptors or eavesdroppers without secret knowledge (key). The sequence of data processing steps required for the transformation of plaintext into cipher text is called message encryption. Encryption algorithms are broadly of two types: private key encryption (e.g. AES, DES) and public key encryption (e.g. RSA).

1.2 Message authentication:

Message authentication allows one party (sender) to send a message to another party (receiver) in such a way that if the message is modified en-route the receiver will almost detect this message. Message authentication(called Data origin authentication)is said to protect the integrity of a message ensuring that each message it has received and accepted is in the same condition that it was sent out with no bits inserted, missing or modified. Authentication techniques are broadly of two types: private key authentication (e.g. MAC) and public key digital signatures (e.g. DSS, ECDSA).

1.3 Digital signature:

A digital signature is a mathematical scheme for ensuring 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, such that the sender cannot deny having sent the message (authentication and non-repudiation) and that the message was not altered in transit (integrity). Digital signatures are commonly used for software distribution, financial transactions, and in other cases where it is important to detect forgery or tampering. The following Figure 1.1 shows the basics of a digital signature scheme [1].

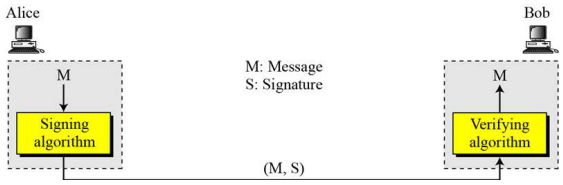


Figure 1.1: Digital Signature Process

Digital signatures cannot provide confidentiality for the message. If confidentiality is needed, a cryptosystem must be applied over digital signature scheme.

1.4 Signature-Then-Encryption:

This is the traditional method to achieve both confidentiality and authenticity, by serial composition of signature and encryption algorithms. This is a two-step approach in which, before a message is sent out, the sender of the message would sign it using a digital signature scheme, and then encrypt the message (and signature) using a private key encryption algorithm under a randomly chosen message encryption key [4]. The random message encryption key would then be encrypted using the recipients public key. The process [6] is shown in the Figure 1.2 . This two-step-approach not much efficient than applying signature and encryption individually.

1.5 Signcryption:

Signcryption is a cryptographic primitive proposed by Yuliang Zheng in 1997, which achieves confidentiality and authenticity in a single logical step. Compared with traditional methods, it has less computational complexity and computational complexity. Signcryption has found many applications such as electronic transactions protocol, mobile agent protocol, key management and routing protocol.

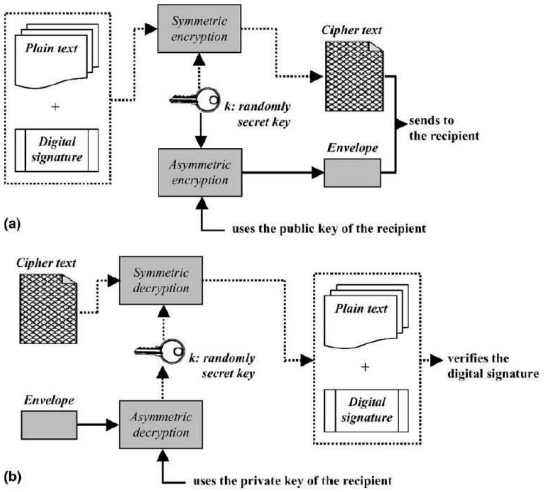


Figure 1.2: (a) Signature-Then-Encryption (b) Decryption-Then-Verification

1.6 Generalized Signcryption:

Generalized Signcryption [7]proposed by Yiliang Han is a cryptographic primitive which can work as an encryption scheme or a signature scheme or a signcryption scheme as per the need. In other words without any additional modification and computation, it provides double functions when confidentiality and authenticity are required simultaneously and the separate encryption or signature function when one of them is required.

A Generalized Signcryption scheme is a two-party cryptographic protocol. The syntax is as follows [8]:

Definition (Generalized Signcryption): A Generalized Signcryption scheme $GSC = (Gen, SC, DSC)$ consists of three algorithms. $(SK_U, PK_U) \leftarrow Gen(U, 1^k)$is a randomized keys generation algorithm, takes a secure parameter k and generates a pair of keys for user U. SK_U is the private key and PK_U is the public key. $\sigma \leftarrow SC(m, SK_S, PK_R)$ is a probabilistic signcryption algorithm. For any message m, the sender S and the receiver R, it outputs a ciphertext σ. $m \cup \perp \leftarrow DSC(\sigma, SK_R, PK_S)$ is a deterministic designcryption algorithm.

Where, $ENC = (Gen, Enc, Dec)$is an encryption scheme. $\varepsilon \leftarrow Enc(m, PK_R).m \leftarrow Dec(\varepsilon, SK_R)$. Enc is an encryption algorithm. Dec is a corresponding decryption algorithm.

It can be clearly observed that, generalized signcryption provides three functionalities (Signcryption, Signature, Encryption) by using a generic primitive instead of switching to several subroutines.

1.7 Thesis Organization

The remainder of the thesis is organized as follows:

Chapter 2 discusses various literature survey works related to the thesis. The Survey has been categorized into four groups of Generalized Signcryption Schemes-Elliptic Curve based Schemes, Bilinear pairing based Schemes, Identity based Schemes and Certificateless Schemes.

Chapter 3 describes about the mathematical preliminaries that are required for the implementation of the proposed scheme. It also discusses the hash functions and Elliptic curve cryptosystems.

Chapter 4 discussed the first proposed scheme named-*a modified Identity based generalized signcryption scheme* in details. The computational and implementational complexities are compared with other existing Signcryption and Generalized Signcryption schemes.

Chapter 5 discussed the second proposed scheme named*an improved Certificateless generalized signcryption schemes* in details. The computational, communicational and implementational complexities are compared with other existing signcryption and generalized signcryption schemes.

Chapter 6 discusses the concluding remarks with the future scope.

Chapter 2

Literature Review

2.1 Related Work

Confidentiality and Integrity are two major requirements in any computer and communication systems. Generally these requirements are achieved by encryption and signature in public key cryptography. Traditionally, these two have been treated as independent entities. However these two basic cryptographic techniques can be combined together in different ways like: sign-then-encrypt, encrypt-then-sign and sign and encrypt etc. in many application to ensure privacy and authenticity simultaneously. The method is used in some famous security protocols like secure sockets layer (SSL), internet Protocol Security (IPsec), and Pretty Good Privacy (PGP). Unfortunately, the method is not practical for two reasons. First, it has low efficiency and cost is sum of the authentication and encryption. Second, not all schemes can guarantee the security. To enhance the efficiency, in 1997 Zheng [3] proposed a novel cryptographic primitive called "Signcryption" which achieves confidentiality and authenticity in a single logical step. Compared with the available traditional methods, signcryption has less computational complexity, less communication complexity and less implementation complexity. Signcryption has found many applications such as electronic transactions protocol, mobile agent protocol, key management and routing protocol, key management and routing protocol. In 2002, Baek

et al. [9] first formalized and defined security notions for signcryption. Many Signcryption schemes have been proposed based on RSA problem [10,11] based on Diffie-Hellman problem libert2004efficient,bao1998signcryption, based on Bilinear Maps [12, 13]. Depending on construction techniques Hybrid Signcryption and parallel Signcryption are also designed. Signcryption schemes for multi-receiver are also designed. Signcryption schemes for multi-receiver are also designed which targets application like broadcast signcryption, Multicast Signcryption etc. some signcryption schemes with additional properties are designed like identity based signcryption [5] and group signcryption [14] and so on.

Though traditional signcryption is efficient to provide both Signature and Encryption functions simultaneously it will not be useful in scenarios where sometimes we need only one function and sometimes we need both simultaneously. In fact, not all messages require both secrecy and authenticity. Some message need to be may need to be signed only, while some others need to be encrypted only. Zheng suggested that signcryption is replaced with other signature or encryption algorithms to resolve the problem. So, applications must contain at least three cryptographic primitive (signcryption, signature, and encryption), which will be in-feasible in some resource-constrained environments like :embedded systems, sensor networks and ubiquitous computing where it will not be affordable to use three different schemes to achieve confidentiality and authenticity separately or simultaneously. Motivated by this, in 2006 Yiliang Han [15] proposed a new primitive called Generalized signcryption, which can work as an encryption scheme or a signature scheme or a Signcryption as need. In other words without any additional modification and computation, it provides double functions when the confidentiality and the authenticity are required simultaneously , and the separate encryption or signature function when one of them is required. The first generalized signcryption scheme named ECGSC (elliptic curve generalized signcryption) is based on ECDLP. Wang et al. [16] gave the formal security model for a Generalized Signcryption scheme and modified the scheme proposed in [15]. Following this many Generalized signcryption schemes has been proposed including some of the standard Generalized Signcryption like IDGSC

(ID based Generalized Signcryption), BPGSC (Bilinear Pairing based Generalized Signcryption), CLGSC (Certificateless Generalized Signcryption) etc. Some of the Generalized Schemes are also proposed for Multiuser model like [5, 17]. Form majority of the schemes, formal security model is based on Random Oracle Model and for few schemes, security model are based on Standard model [18].

2.2 Identity based cryptosystems:

Identity based cryptosystem was introduced by Shamir in 1984 [19]. The central idea here is to use any string as a public key. In particular this string may be the email address, telephone number, social security number or any publicly available parameter of a user that is unique to him. The corresponding private key can only be derived by a trusted Private Key Generator(PKG) which keeps a master secret that is use to derive the private keys. So this greatly relieves the burden of public key management and provides a more convenient alternative to Public Key Infrastructure(PKI). The major disadvantage of Identity based cryptosystem is the key escrow problem [2], where a untrusted PKG will have the power to forge signatures in the name of any user of the system, as well as the ability to decrypt all of their private communications.

2.3 Certificateless Cryptography

Certificateless Cryptography was introduced by Al-Riyami and Paterson [20] that avoids drawbacks of both traditional PKI and Identity based cryptosystems and acheives the best of both: it inherits from identity-based techniques a solution to the certificate management problem and also removes the secret key escrow problem inherent to the identity based systems. The idea was to combine the functionality of a public key scheme with that of an identity based scheme. User encryption and verification keys contain both a user identity and an unauthenticated public key. Similarly, user secret keys are constructed from two partial secrets: one coming from an identity-based trusted

authority called Key Generation Center(KGC) and another generated by the user.

2.4 Framework of ID based generalized Signcryption Schemes

The Algorithm for a generic Identity Based Signcryption scheme IDGSC = (Setup, KeyGenration, GSC, GUSC) consists of four algorithms which are:

- **Setup** (1^k)**:** This is a randomized algorithm run by **PKG**. Given a security parameter k, this algorithm generates the system parameters *params* and master secret key s and master public key *mpk*.
- **KeyGenration** (mpk, msk, ID)**:** On input ID, PKG uses it to compute a pair of corresponding public/private keys (S_U, Q_U).
- **GSC:** To send a message m from Sender S to the Receiver R, this algorithm takes input (S_S, ID_R, m) and outputs signcrypted text $\sigma = MIDGSC(S_S, ID_R, m)$.
  - When $ID_S \neq ID_\phi, ID_R \neq ID_\phi, \sigma \leftarrow GSC(S_S, Q_R, m) = SC(S_S, Q_R, m)$
  - When $ID_S \neq ID_\phi, ID_R = ID_\phi, \sigma \leftarrow GSC(S_S, Q_R, m) = Sign(S_S, m)$
  - When $ID_S = ID_\phi, ID_R \neq ID_\phi, \sigma \leftarrow GSC(S_S, Q_R, m) = Encrypt(Q_R, m)$
- **GUSC:** This algorithm takes input (ID_S, S_R, σ) and outputs m if σ is a valid Generalized Signcryption done by Sender S for Receiver R, otherwise output false $(\perp)$ if is not valid.
  - When $ID_S \neq ID_\phi, ID_R \neq ID_\phi, m \leftarrow GUSC(Q_S, S_R, \delta) = USC(Q_S, S_R, \delta)$

- When $ID_S \neq ID_\phi, ID_R = ID_\phi, (T, \perp) \leftarrow GUSC(Q_S, S_R, \delta) = Verify(S_S, \delta)$
- When $ID_S = ID_\phi, ID_R \neq ID_\phi, m \leftarrow GUSC(Q_S, S_R, \delta) = Decrypt(Q_R, \delta)$

The absence of specific sender or receiver are denoted by ID_ϕ, ID_ϕ instead of ID_S, ID_R. When there is no specific sender(ID_ϕ) we only encrypt the message m using MIDGSC, when information about sender is not needed MIDGSC becomes signature scheme and when both are there it will work as Signcryption scheme.

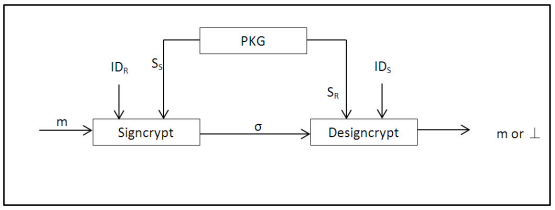


Figure 2.1: Identity Based Signcryption

2.5 Framework Certificateless Generalized Signcryption Schemes

This scheme consists of six algorithms. First four of which are used for key management operations.

1. **Setup**(1^k)**:**This is a global setup algorithm, which takes input the security parameter 1^k and returns the **KGC**'s secret key **msk** and global parameters params including a master public key **mpk**. This algorithm is executed by the **KGC**, which publishes *params*.

2. **Extract-partial-private-key** ($ID_U, msk, params$)**:** An algorithm which takes input *msk*, *params* and a user identity $ID_U \in \{0,1\}^*$ and returns a partial private key D_U. This algorithm is run by **KGC**, after verifying the users identity.

3. **Generate-User-Keys ($ID_U, params$):** An algorithm which takes input as an identity and the public parameters and outputs a secret value x and a public key PK. This algorithm is run by a user to obtain a public key and a secret value which will be used for constructing full private key. The public key is published without certification.

4. **Set-Private-Key ($D_U, x, params$):** A deterministic algorithm which takes as input a partial secret key D_U and a secret value x and returns the full private key S_U. This algorithm is run by a user to construct a full private key.

5. **CLGSC (m, S_S, ID_R):** This algorithm has three scenarios: signcryption mode, signature only mode and encryption only mode.

   - **Signcryption Mode:** If sender S transmits wants to transmit a message m to receiver B such that both confidentially and authentication need to be maintained then the input is (m, S_S, ID_R), and output is $\sigma = CLGSC(m, S_S, ID_R) = Singcrypt(m, S_S, ID_R)$.
   - **Signature only Mode:** If sender S wants to send message m without definite receiver, the input is (m, S_S, ID_ϕ), where ID_ϕ means receiver is null, the output is$\sigma = CLGSC(m, S_S, ID_\phi) = sign(m, S_S)$.
   - **Encryption only Mode:** If someone wants to send a message m to a definite receiver R confidentially, the input is (m, S_ϕ, ID_R), where S_ϕ means the receiver is null, the output is$\sigma = encrypt(m, ID_R)$.

6. **CLGDSC (σ, ID_S, ID_R):** After receiving σ, if it is valid, the receiver R designcrypts (or decrypts) the ciphertext and returns the message m and (or) the signature on m by S, otherwise return ($\perp$) means false.

2.6 Comparison of Existing Generalized Signcryption Schemes:

The works on Generalized Signcryption till date can be broadly classified into be four categories like:

1. **Elliptic Curve Based Schemes**
    - ECGSC:Elliptic Curve-Based Generalized Signcryption by Yiliang Han [7] in 2006.
    - Generalized Signcryption Scheme Based on short ECDSA by Zhang *et al.* [21] in 2010.
    - Provable Secure Generalized Signcryption Scheme by Wang *et al.* [16] in 2010. security model by Han is not correct, and proposed the formal security model.
2. **Bilinear Pairing Based Schemes**
    - BPGSC: Bilinear Pairing based Generalized Signcryption by Han *et al.* [8]in 2009. Also proposed hybrid BPGSC to transmit large data streams.
3. **ID Based Schemes**
    - IDGSC : ID based Generalized Signcryption by Lal *et al.* [22] in 2008 , based on Boneh-Franklin ID-based encryption.
    - Generalization of Barreto *et al.* ID based Signcryption Scheme by Lal *et al.* [23]in 2008.
    - NIDGSC: Provable Secure ID based Generalized Signcryption Scheme by Yu *et al.* [24]in 2010.
    - IBGSC: An efficient Identity Based Generalized Signcryption Scheme by Kushwah *et al.* [25]in 2011.
4. **Certificateless environment**

- CLGSC: Certificateless Generalized Signcryption by Ji *et al.* [26] in 2010.
- Efficient Generalized Signcryption Scheme by Kushwah *et al.* [27] in 2010.
- Provable Secure Certificateless Generalized Signcryption Scheme by Kushwah *et al.* in 2012.
- N-CLGSC Provable Certificateless Generalized Signcryption Scheme by Zhou *et al.* [28] in 2012.

The following Table 2.1 shows the comparison of Elliptic curve based Generalized Signcryption Schemes with other existing Signcryption schemes.

Table 2.1: Comparison of Computational Cost

Schemes	KG	S	D
SCS	2E	1E+1I	2E
ECSCS	2kP	1kP+1I	2kP
B&D	2E	2E+1I	3E
SC-DSA	2E	2E+2I	3E+1I
ECGSC	2kP	2kP+1I	3kP+1I

The Table 2.2 below shows the comparison of ID based and Certificateless signcryption schemes with other existing signcryption schemes.

M: number of point multiplications in G_1; E: number of exponentiation in G_2; P:number of pairing computations; (+): pre-computation of pairing.

2.7 Observation

The Generalized Signcryption schemes are capable of providing multiple functionalities with comparable cost as compared to the normal Signcryption schemes. Some times the cost is bit more than signcryption schemes that is because of the additional functionalities that they provides.

Table 2.2: Comparison of Computational Cost

Schemes	Signcryption			UnSigncryption		
	M	E	P	M	E	P
Malone Lee	3	0	0(+)	1	0	3(+1)
Nalla-Ready	2	1	1	0	1	3
Libert-Quisquater	2	0	0(+2)	1	0	4
X. Boyen	3	1	0(+1)	2	0	3(+1)
IDGSC	5	0	0(+1)	1	0	3(+1)
NIDGSC	3	1	0(+1)	0	2	2(+2)
Bareto based GSC	2	1	0	0	1	2
IBGSC	2	2	0	1 or 0	1 or 0	2 or 1
CLGSC	3	2	0	1	1	2
Efficient CLGSC	2	3	0	1	3	2
NCLGSC	1	4	0(+1)	0	1	4(+1)

2.8 Motivation

Encryption and Signature are fundamental tools in public key environment for providing confidentiality and authenticity respectively. Traditionally they are provided by Sequential composition. When applications need both functionalities simultaneously, Solution is Signcryption. Applications may need the two functionalities Simultaneously or may be separately without increasing complexity. Specifically in Resource constrained environments such as Sensor networks,Mobile Computing, and smart card based applications cannot afford separate modules for achieving both functionalities.

2.9 Objective of Research

1. To design a Multi functional Generalized Signcryption in Identity Based Cryptosytems for application in Resourse constarined environment.

2. To design a Multi functional Generalized Signcryption in certificateless environment to avoid Key Escrow problem.

Chapter 3

Mathematical Background

3.1 Mathematics of Cryptology

The basic properties of modular arithmetic, groups, rings, fields, fundamentals of elliptic curves, Bilinear mappings, cryptographic hash functions are discussed in this Chapter.

3.1.1 Modular Arithmetic

Set of Residues: Z_n

The result of the modulo operation with modulus n is always an integer between 0 and n-1. The modulo operation creates a set, which in modular arithmetic is referred to as the set of least residue moduli n, Z_n. The Set Z_n and its 3 instances are shown below

Z_n={0, 1, 2, .(n-1)}

Z_2= {0, 1},

Z_6= {0, 1, 2, 3, 4, 5},

Z_{11}= {0 1 , 2, 3 , 4, 5, 6, 7, 8, 9, 10}

Additive Inverses

In Z_n, two numbers a and b are additive inverses of each other if $a + b \equiv 0(mod n)$.

In Z_n additive inverse of a can be calculated as $b = n - a$. For example , the additive inverse of 4 in Z_n is $10 - 4 = 6$.

Multiplicative Inverse

In Z_n, two numbers are multiplicative inverses of each other if $a \times b \equiv 1(mod n)$. For example in Z_n, the multiplicative inverse of 3 is 7.

The integer a inZ_n has a multiplicative inverse if and only if $gcd(n, a) \equiv 1(mod n)$. In this case, a and n are said to be relatively prime. For example, there is no multiplicative inverse of 8 in Z_{10} because $gcd(8, 10) = 2 \neq 1$.

Some new Sets

1. $Z_n{}^*$: The set,$Z_n{}^*$ is a subset of Z_n and includes only integers in Z_n that have a unique multiplicative inverse.

   Each member Z_n has an additive inverse, but only some members have a multiplicative inverse. Each member of $Z_n{}^*$ has a multiplicative inverse, but only some members have a multiplicative inverse.

   Example:

   Z_6= {0, 1, 2, 3, 4, 5} $Z_6{}^*$= {1, 5}

   Z_7= {0, 1, 2, 3, 4, 5, 6} $Z_7{}^*$= {1, 2, 3, 4, 5, 6}

   Z_{10}= {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} $Z_{10}{}^*$= {1, 3, 7, 9}

2. Z_p: The set Z_p is same as Z_n except that n is a prime. Z_p contains all integers from 0 to p-1. Each member in Z_p has an additive inverse; each member except 0 has multiplicative inverse. Note: We need to use Z_n when additive inverses are needed; we need to use $Z_n{}^*$ when multiplicative inverses are needed.

3. $Z_p{}^*$: The Set $Z_p{}^*$ is same as $Z_n{}^*$ except that n is prime. $Z_p{}^*$ contains all integers from 1 to p-1. Each member in $Z_p{}^*$ has an additive and multiplicative inverse. $Z_p{}^*$ is a very good candidate when we need a set that supports both additive and multiplicative inverse.

   Example:

Z_{13}= {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}
$Z_{13}{}^{*}$= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}

3.1.2 Mathematics of Symmetric key Cryptography

Algebraic Structures:

Cryptography requires sets of integers and specific operations that are defined for those sets. The combination of the set and the operations that are applied to the elements of the set is called an algebraic structure. Some of the common algebraic structures: groups, rings, and fields.

Grouos: A group G is a set of elements with a binary operation "$\bullet$" that satisfies four properties .

1. Closure: if x and y are the elements of G, then $z = x \bullet y$ is also an element of G.

2. Associativity: If x, y and z are elements of G, then $(x \bullet y) \bullet z = x \bullet (y \bullet z)$.

3. Existence of identity: For all x in G, there exist an element e, called the Identity element, such that $e \bullet x = x \bullet e = x$.

4. Existence of inverse: For each x in G, there exist an element x, called the inverse of x, such that $x \bullet x' = x' \bullet x = e$.

Along with those properties if it also satisfies the commutative property then it is called as Commutative group or Abelian group. Commutativity means for all x and y in G, we have $x \bullet y = y \bullet x$

Order of a Group: The order of a group, $|G|$, is the number of elements in the group. If the group is not finite, its order is infinite; if the group is finite, the order is finite.

Subgroups: A Subset H of a group G is a subgroup of G if H itself is a group with respect to the operation on G. In other words, if $G =< S, \bullet >$ is a group, $H =< T, \bullet >$ is a group under the same operation, and T is a nonempty subset of S, then H is a subgroup of G. The above definition implies that:

1. If x and y are the members of both groups, then $z = x \bullet y$ is also a member of both groups.

2. The groups share the same identity element.

3. If x is a member of both groups, the inverse of x is also a member of both groups.

4. The group made of the identity element of $G, H =< e, \bullet >$, is a subgroup of G.

5. Each group is a subgroup of itself.

Cyclic Subgroups: If a subgroup of a group can be generated using the power of an element, the subgroup is called the cyclic subgroup. The term power here means repeatedly applying the group operation to the element: $x^n = x \bullet x \bullet x \bullet .....(n\,times)$ The set made from this process is referred to as $< a >$.

Cyclic Groups: A cyclic group is a group that is its own cyclic subgroup. The element that generates the cyclic subgroup can also generate the group itself. This element is referred to as a generator. If g is a generator the, the elements in a finite cyclic group can be written as $\{e, g, g^2, ...g^{n-1}\}$, where $g^n = e$. A cyclic group can have many generators. Example: The group $G =< Z_6, + >$ is a cyclic group with two generators, $g = 1$ and $g = 5$ The group $G =< Z_{10}, \times >$ is a cyclic group with two generators, g=3 and g=7.

Lagrange's Theorem: The order of a subgroup ($|H|$) divides the order of the group ($|G|$). This implies the number of subgroup of a group can be easily determined by the divisors of order of a group ($|G|$). Given, the order of the group $G =< Z_{17}, + >$ is 17. The only divisors of 17 are 1 and 17. This means this group can have two subgroups, H_1 with the identity element and $H_2 = G$.

Order of an Element: The order of an element a in a group, $ord(a)$, is the smallest integer n such that $a^n = e$. This also implies that, the order of an element is the order of the cyclic group it generates.

Ring: A ring denoted as $R =< ..., \bullet, \square >$, is an algebraic structure with two operations. The first operation must satisfy all five properties required for an

abelian group. The second operation must satisfy only first two properties (Closure and Associativity). In addition the second operation must be distributed over the first. Distributivity means that for all x, y and z elements of R, we have $x\Box(y \bullet z) = (x\Box y) \bullet (x\Box z)$ and $(x \bullet y)\Box z = (x\Box z) \bullet (y\Box z)$.

A ring is said to be a **commutative ring** if the second operation also satisfies the Commutativity property.

Field: A Field, denoted by $F =< \{...\}, \bullet, \Box >$ is a commutative ring in which the second operation satisfies all five properties defined for the first operation except that the identity of the first operation(*zero element*) has no inverse.

Finite Fields: Only finite fields are extensively used in cryptography. Galois showed that for a field to be finite the number of elements should be p^n, where p is a prime and n is a positive integer. The finite fields are usually called **Galois fields** and denoted as $GF(p^n)$.

A Galois Field, $GF(P^n)$, is a finite field with p^n elements.

G$F(p)$ **Fields:** When n=1, we have $GF(p)$ field. This field can be the set $Z_p = \{0, 1, 2, ...p-1\}$, with two arithmetic operations , addition and multiplication.

G$F(2^n)$ **Fields:** When we work with computers, the positive integers are stored in the computer as n-bit words in which n is usually 8, 16, 32, 64, and so on. This means the range of integers is 0 to 2^n and the modulus is 2^n. The elements in the set are n bit words. For example for n=3, the set is $\{000, 001, ...111\}$.

3.2 Elliptic Curve Cryptosystem

Elliptic Curve (EC) systems as applied to cryptography was applied to cryptography was first proposed in 1985 independently by Neal Koblitz and Victor Miller. The Elliptic Curve Cryptosystem is based on the theory of Elliptic Curves.

3.2.1 Definition of Elliptic Curves

An Elliptic curve over a finite field K is a non-singular cubic curve in two variables, $f(x, y) = 0$ with a rational point (which may be a point at infinity). The field K

is usually taken to be the complex numbers, reals, rationals, algebraic extensions of rationals, p-adic numbers or a finite field. By, non-singular means all 3 roots of EC must be distinct roots*nodoubleroots*.

3.2.2 General form of an EC:

An elliptic curve is a plane curve defined by an equation of the form $y^2 = x^3+ax+b$ Here x is not a continuous point, chosen from a particular field $GF(P)$ or $GF(2^n)$.

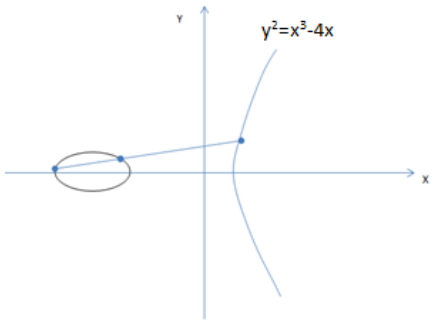


Figure 3.1: Graphical representation of an elliptic curve

- Symmetric over X-axis
- Cubic curve in the variable x
- Although we are actually drawing continuous elliptic curves they are actually discrete curves (Discrete collection of points).

3.2.3 Why ECC?

One of the main problems of RSA is its demand for a huge key length to meet the challenges in today's security scenario. Also in every 10 years key size becomes double. Table 3.1 shows some of the currently used RSA key lengths by some organizations. A larger key increases the security of the encryption. But it has a serious problem in practice. With every doubling of the RSA key length, decryption is about 8 times slower. The size of ciphertext also becomes

Table 3.1: RSA key length of some organizations

Organization	RSA Key length
Google	1024
Facebook	1024
Amazon	2048
eBay	2048
Online SBI	2048
ICICI Bank	2048
Canara Bank	2048

huge considerably. The key length also affects the speed of encryption, which is slower by a factor of 4. The comparisons in Table 3.2 demonstrate that smaller parameters can be used in elliptic curve cryptography (ECC) than with RSA system at a given security level.

Table 3.2: RSA and ECC key sizes

Security Level	80	112	120	128	256
ECC	160	185	237	256	512
RSA	1024	2048	2560	3072	15360

The advantages that can be gained from smaller parameters include speed (faster computations) and smaller keys and certificates.

3.2.4 Weierstrass Equation

Elliptic curves are a specific class of algebraic curves. Common or more generalized form of the elliptic curve equation is known as Weierstrass Equation. The 'Weierstrass form' of an elliptic curve equation is $E : y^2 + a_1xy + a_3 = x^3 + a_2x^2 + a_4x + a_6$ The Constants $a_1, a_2, a_3, a_4, a_6$and the variables x, y can be complex, real, integers, polynomials, or even any other field elements. But in practice we must specify which field, F, these constants

and the variables, x, y belong to and $\Delta \neq 0$ whereΔ is the discriminant of E and is defined as follows: $\Delta = -d_2^2 d_8 - 8d_4^3 - 27d_6^2 + 9d_2 d_4 d_6$ $d_2 = a_1^2 + 4a_2$ $d_4 = 2a_4 + a_1 a_3$ $d_6 = a_3^2 + 4a_6$ $d_8 = a_1^2 a_6 + 4a_2 a_6 - a_1 a_3 a_4 + a_2 a_3^2 - a_4^2$ E is defined over K when the coefficients a_1, a_2, a_3, a_4, a_6 (also the variables x and y) of the equations come from the elements of the field K. So sometimes it can be written E (K) to emphasize that E is defined over K, and K is called the underlying field. Two special Galois fields are common in Elliptic Curve Cryptography. They are $GF(p)$ and $GF(2^n)$.

3.2.5 Elliptic Curve Over Prime Galois Fields

An elliptic group over a prime Galois Field uses a special elliptic curve of the form$y^2(mod p) = x^3 + ax + b(mod p)$ Where $a, b \in GF(p)$,$0 \leq x \leq p\, and\, -16(4a^3 + 27b^2) mod\, p \neq 0$. The constants **a** and **b** are non-negative integers smaller than the prime p. The condition that $-16(4a^3 + 27b^2) mod p \neq 0$ implies that the curve has no "singular points".

3.2.6 Group Law

The mathematical property that makes elliptic curves useful for cryptography is simply that if we take two (distinct) points on the curve, then the chord joining them intercepts the curve in a third point (because we have a cubic curve). If that point is reflected in the x-axis we get another point on the curve (since the curve is symmetric about the x-axis). This is the "sum" of the first two points. Together with this addition operation, the set of points E (K) forms an abelian group with O serving as its identity. It is this group that is used in the construction of elliptic curve cryptographic systems. Algebraic formula for the group law can be derived from the geometric description. Group law for $y^2 = x^3 + ax + b$ over $GF(p)$

1. Identity: $P + O = O + P = P$ for all $P \in E(K)$.

2. Negative: If $P = (x, y) \in E(K)$, then $(x, y) + (x, -y) = O$. The point

$(x, -y)$ is denoted by $-P$ and is called the negative of P; note that $-P$ is indeed a point in $E(K)$. Also,$-O = O$.

3. Point addition: Let $P(x_1, y_1) \in E(K)$ and $Q(x_2, y_2) \in E(K)$ where $P \neq Q$. Then $P + Q = R(x_3, y_3)$, where $x_3 = \lambda^2 - x_1 - x_2$,$y_3 = \lambda(x_1 - x_3) - y_1$ and $\lambda = (y_2 - y_1)/(x_2 - x_1)$.

4. Point doubling: Let $P(x_1, y_1) \in E(K)$, where $P \neq P$. Then $2P = R(x_3, y_3)$ where $x_3 = \lambda^2 - 2x_1$, $y_3 = \lambda(x_1 - x_3) - y_1$ and $\lambda = ((3x_1^2 + a))/(2y_1)$

3.2.7 Geometrical Interpretation of Group Law

1. Negative of a Point: Let's take a point $P = (x, y)$. The formula for finding -P is $-P = (x, -y)$ as shown in the fig. 1.

2. Addition of two Points: As mentioned above, the addition of any two points on an elliptic curve can defined by drawing a line between the two points and finding the point at which the line intersects the curve. The negative of the intersection point is defined as the "elliptic sum" by mathematicians as shown in fig. 2.

3. Doubling of a Point: If $P(x_1, y_1)$, then the double of P, denoted by, $R(x_3, y_3)$, is defined as follows. First draw the Tangent line to the elliptic curve at P. This line intersects the elliptic curve in a second point. Then R is the reflection of this point in the x-axis.

ECC in cryptography: Like RSA has exponentiation, ECC has point multiplication (repeated addition of two points) as its underlying mathematical operation. Scalar Multiplication: Scalar λ, Base point P then $\lambda P = P + P + P + .. + P(\lambda times)$. A base point B is taken from the elliptic group (similar to generators used in other cryptosystems). Private Key: an integer x, selected from the interval $[1, p - 1]$. Public key: $Q = x \times B$. Hard Problem of ECC: It is analogous to discrete log problem. Let $Q = kP$, where P, Q are points on elliptic curve.

Given $k, P =>$ "easy"to compute Q.
Given $Q, P =>$ "hard" to find K. this is known as elliptic curve discrete logarithmic problem.

Discrete Logarithmic problem: If g and h are elements of a finite cyclic group G then a solution x of the equation $g^x = h$ is called discrete logarithm to the base g of h in the group G.
Elliptic Curve Discrete Log Problem (ECDLP) Let E be an elliptic Curve over the finite field F_p. And let P, Q be the points in $E(F_p)$. The Elliptic Curve Discrete Logarithm Problem (ECDLP) is the problem of finding n such that $Q = nP$. The integer n is denoted as $n = log_p(Q)$, the elliptic curve discrete logarithm of Q with respect to P.

3.2.8 Applications of ECC:

ECC is ideal for devices which are small and have limited storage and computational power. Like

- Wireless communication devices
- Smart cards
- Online transactions
- Web Servers
- Any application where security is needed but lacks the power, storage and computational power that is necessary for present day applications.

3.3 Cryptographic Hash Function

3.3.1 Hash Function:

A cryptographic hash function is a deterministic function which maps a string of arbitrary length to a string of fixed length called hashed value (sometimes called

message digest).

$$h : \{0,1\}^* \rightarrow \{0,1\}^n$$

Consider a function $f(x) = y$ that maps x to the image y. The x is said to be the preimage of y.
The output is called hash value or message digest or simply digest. Here, we assume $y = h(x)$, where h is a (public domain) hash function, which has to meet the following requirements:

1. The length of x is arbitrary, while the length of y is constant;
2. For a given x, it is easy to compute y; while, for a given hash value y, it is hard to find x to satisfy $y = h(x)$;
3. It is computationally unfeasible to find two different inputs x and x and $h(x) = h(x)$.

Cryptographic hash functions are used universally in cryptography; digital signatures, message authentication codes (MAC), random sequence generators used in key agreement, authentication protocols etc.
Cryptographic hash functions need to satisfy the following three security properties:

1. Preimage Resistance: Given a digest y, it is computationally infeasible to find a message x that hashes to y. That is, computational cost of finding the input x must be $\geq 2^n$, where $h(x) = y$ and $|y| = n$.

Instance: A hash function $h : X \rightarrow Y$ and an element $y \in Y$.
Find: $x \in X$ such that $h(x) = y$.

If the preimage can be solved then (x, y) is a valid pair. A hash function for which preimage cannot be efficiently solved is said to be preimage resistant.

2. Second Preimage Resistance: Given a message x, it is computationally infeasible to find a different message x', such that both messages hash to a same digest. That is, computational cost of finding the input$x'(\neq x)$ must be $\geq 2^n$, where $h(x') = y, h(x) = y$, and $|y| = n$.

Instance: A hash function $h : X \to Y$ and an element $x \in X$.
Find: $x' \in X$ such that $x' \neq x$ and $h(x') = h(x)$.

   If the second preimage problem is solved then, the pair $(x', h(x))$ is valid. If it cannot be done efficiently then the hash function is second preimage resistant.

3. Collision Resistance: It is computationally infeasible to find two different messages, which hash to the same digest. That is, computational cost of finding an input pair$x$ and x' such that $h(x) = h(x')$. Here n the length of the message digest.

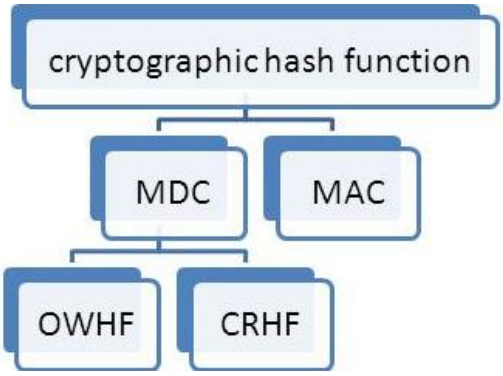
Instance: A hash function $h : X \to Y$.
Find: $x, x' \in X$ such that $x' \neq x$ and $h(x') = h(x)$.

   If this collision problem can be solved then if (x, y) is valid pair so is (x, y). If not efficiently solvable the hash function is called collision resistant.

From attacker point of view Collision Resistance is a simple problem, but from designers point of view, it is a much harder problem.
The preimage resistance property can be expressed as the inability to learn about the contents of the input data from its digest. The second preimage resistance property can be interpreted as the inability to learn about the second preimage from the given first preimage such that both of these preimages have the same digest. The collision resistance property signifies that the digests are almost unique for each given message. If the input message is altered, almost always the hash changes as well. The word almost is used, because when a function maps from a larger domain to a smaller range, collisions necessarily exist. If cryptographic

hash functions are designed properly, with digests of sufficient length then the probability that one can obtain two different messages with identical hashes is too small to be bothered in all practical applications.

These three properties preimage resistance, second preimage resistance and collision resistance are also known as one-way, weak collision resistance, and strong collision resistance properties respectively. If a hash function satisfies the first two properties then it is referred as one-way hash function (OWHF). Whereas the hash function that satisfies all the three properties referred as collision resistant hash function (CRHF). A hash function with an output of n bits can only offer a security level of 2^n operations for pre-image and second pre-image attacks and $2^{n/2}$ operations against finding collisions. Apart from these properties, it



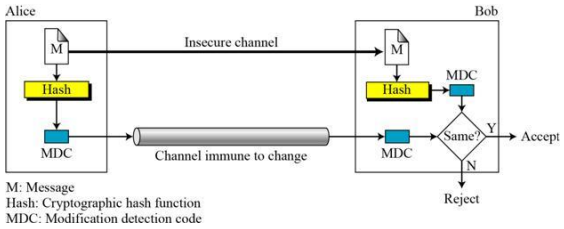
is expected that a good hash function will satisfy some properties (desirable but not necessary), they are

- Near-collision resistance: A hash function is said to be near-collision resistant if it is hard to find any two messages x and x' such that $x \neq x'$ and $h(x) \oplus h(x') = \Delta$ for small difference Δ.

- Partial-preimage resistance: A hash function satisfies this property when the difficulty of finding a partial preimage for a given digest is the same as that of finding a full preimage using digest. It must also be hard to recover the whole input even when part of the input is known along with the digest.

A message digest guarantees the integrity of a message. It guarantees that the message has not been changed. A message digest however does not authenticate

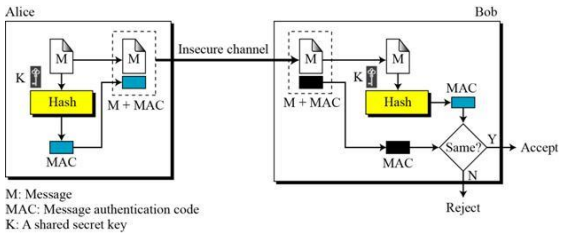
the sender of the message. The digest created by cryptographic hash function is normally called a modification detection code (MDC). The code can detect any modification in the message. There is another type of hash that uses a secret key that is MAC (message authentication code). Conditions:



- The input X can be of arbitrary length and the result$h(X, K)$ has a fixed length of n bits. The function has an secondary input the key K, with a fixed length of k bits.
- Given h, K and an input X, the computation of $h(K, X)$ must be easy.
- Given a message X (but with unknown K), it must be 'hard' to determine $h(K, X)$.

3.3.2 Message Authentication Codes(MAC)

MAC is a keyed hash function, used to verify the integrity and authentication of information. A MAC algorithm take a secret key K of length k and an arbitrary length message x as input and returns the authentication tag defined as $MAC(K, x) = MAC_k(x)$. Given a MAC algorithm MAC and the inputs x and K, the computation of tag $MAC_k(x) = \tau$ of fixed size n must be easy. Some of the



cryptographic hash functions use compression function from the scratch. Some of them are described as follows

- Message Digest: Several hash algorithms are designed by Ron Rivest. These are referred as MD2, MD4, and MD5 . The MD5 is the strengthened version of MD4 that divides the message into blocks of 512 bits and creates a 128-bit digest. As 128-bit is too small to resist collision attacks so better to go for Secure hash Algorithms (SHA).

- Secure Hash Algorithm (SHA): The Secure Hash Algorithm is a standard was developed by NIST and was published as a FIP standard. Its mostly based on MD5. The standard was revised in 1995, which includes SHA-1. It is then revised to four new versions: SHA-224, SHA-256, SHA-384, and SHA-512. Characteristics of various SHA are shown in Table 3.3.

Table 3.3: Characteristics of secure hash algorithms

Characteristics	SHA-1	SHA-224	SHA-256	SHA-384	SHA-512
Maximum Message Size	$2^{64}-1$	$2^{64}-1$	$2^{64}-1$	$2^{128}-1$	$2^{128}-1$
Block size	512	512	512	1024	1024
Message Digest Size	160	224	256	384	512
Number of Rounds	80	64	64	80	80
Word Size	32	32	32	64	64

3.3.3 Random Oracle Model:

The Random Oracle Model was introduced by Bellare and Rogaway in 1993, is an ideal mathematical model for a hash function. A function based on this model behaves as follows.

- When a new message of any length is given, the oracle creates and gives a fixed-length message digest that is a random string of zeros and ones. The oracle records the message and the message digest.

- When a message is given for which a digest exists, the oracle simply gives the digest in the record.

- The digest for a new message needs to be chosen independently from all previous digests.

It states that the knowledge of the previously computed values does not give any advantage to the future computations of $h(x)$.

RO is a theoretical model that captures the concept of an ideal hash function. Random Oracle, models the ideal hash function in a way, that is if you access a value of say x and which you have never accessed before then you are returning a random number but if you are accessing something which you have accessed before then you are returning the same number which you have returned before. If a hash function, h is ideal then the only way to compute the hash of a given value is by actually computing it, i.e. even if many previous values are known then also computing the hash of a new value should not be derivable from the previous ones.

Non-Ideal hash function:

The new hash values can be computed from pre-computed values, like linear kind of equation (if you know two points from there you can calculate all the points). Consider a hash function $h : Z_n \rightarrow Z_n$ which is a linear function, say $h(x, y) = ax + bymodn,\ a, b \in Z_n, n \geq 2$is a positive integer. Suppose $h(x_1, y_1) = ax_1 + by_1, h(x_2, y_2) = ax_2 + by_2$.

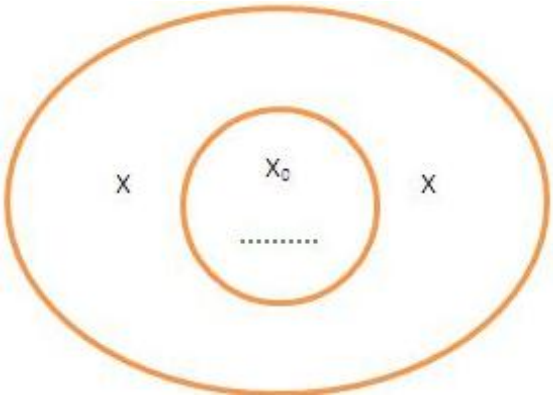
$h(rx_1 + sx_2modn, ry_1 + sy_2modn) = rh_1(x_1, y_1) + sh_2(x_2, y_2)modn$, where $h_1()$ and $h_2()$ are known. Thus we can compute the hash of another hash value apart from (x_1, y_1) and (x_2, y_2) without actually computing the hash value, means we are computing the new hash values from pre-computed values even without knowing the values of a and b. So, these types of hash function are not ideal hash functions according to the **RO** model.

The **RO** model should be such that although suppose we have got a domain say X and out of them suppose we take any subset $X_0$and suppose we know all the corresponding hashed output for this X_0.

Now if you access a point from $X \backslash X_0$ (X difference X_0), then this hashed outputs (dotted in figure) should not give you any information about the other hashed

outputs.

That means before we started any computation if the probability of any hash output occurring is $1/M$ (M is the range), but even after computing X_0, the probability still stays $1/M$, we are not able to compute with any better probability. It states that the knowledge of the previously computed values does not give any



advantage to the future computations of $h(x)$.

3.3.4 Pairing-Based Cryptography:

The central idea behind Pairing-Based Cryptography is the mapping between two useful cryptographic groups which allows new cryptographic schemes based on the reduction of one problem in one group to a different, usually easier problem in the other group.

Bilinear Maps:

Let G_1 be a cyclic additive group generated by P.

Bilinear-Maps are the tool of pairing-based cryptography. They basically establish relationship between cryptographic groups. Bilinear maps are called pairings because they associate pairs of elements from G_1 and G_2 with elements in G_t. This definition admits degenerate maps which map everything to the identity of G_t.

Let G_1, G_2 and G_t be cyclic groups of large prime order q. Generally we write

G_1, G_2 additively and G_t multiplicatively. A pairing is a mapping $e : G_1 \times G_2 \to G_t$, satisfying the property of Bilinearity, which means the following should hold:
$e(aP, bR) = e(P, Q)ab$, for all $P \in G_1, Q \in G_2$ and all $a, b \in Z$.
A pairing is admissible if the mapping is also non-degenerate and computable. Admissible Bilinear Mapping are denoted as. These are the only bilinear maps used in cryptography . Non-degeneracy means mapping cannot be the trivial map which sends every pair of elements of G_1 and G_2 to the identity element of G_t. Because all are groups of prime order, it follows that if P is a generator of G_1 and q is a generator of G_2, then $e(P, Q)$ is a generator of G_t. A mapping is said to be computable if an algorithm exists which can efficiently compute $e(P, Q)$ for any $P, Q \in G_1$. If $G_1 = G_2$ then the pairing is said to be symmetric. Otherwise it is said to be asymmetric. If $G_1 = G_2 = G_t$ then the map is called self-bilinear map $(G \times G \to G)$.
G_1, G_2 and G_t are all isomorphic to one another since they have the same order and are cyclic.
$e : G_1 \times G_2 \to G_t$ such that for all $u \in G_1, v \in G_2, a, b \in Z, e(ua, vb) = e(u, v)ab$.
The Other Notation: Sometimes G is written multiplicatively. In this case P, Q normal names for elements of G Bilinear property expressed as $\forall P, Q \in G$, $\forall a, b \in Z, e(Pa, Qb) = e(P, Q)^{ab}$.

Bilinear Pairing:

Let G_1 be a cyclic additive group and G_2 be a cyclic multiplicative group of the same prime order q. Let P be an arbitrary generator of G_1 and a, b be the elements of Z_q^*. A bilinear pairing is a map.
A bilinear pairing is a map $e : G_1 \times G_1 \to G_2$ with the following properties:

1. **Bilinearity :** for every $P, Q, R \in G_1$, we have $e(P, Q + R) = e(P, Q)e(P, R), e(P + Q, R) = e(P, R)e(Q, R)$
   Consecutively, for any $a, b \in Z_q^*$:
   $e(aP, bQ) = e(P, Q)^{ab} = e(abP, Q) = e(P, abQ) = e(bP, Q)^a$
   $e(kP, Q) = e(P, kQ) = e(P, Q)^k$

2. **Non-Degeneracy:** If everything maps to the identity, thats obviously not desirable. If P is a generator of G_1, then $e(P, P)$ is a generator of G_2. In other words there exist $P \in G_1$ such that $e(P, P) \neq 1$ where 1 is the identity element of G_2.

3. **Computability:** There exist an efficient algorithm to compute $e(P, Q)$ for every $P, Q \in G_1$.

The pairing map e is sometimes called an admissible pairing. A pairing is admissible if the mapping is also non-degenerate and computable($\widehat{e}$).

Definition: Let $\widehat{e} : G_1 \times G_2 \rightarrow G_t$ be a bilinear map. Let g_1, g_2 be two generators of G_1, G_2 respectively. The map e is an admissible bilinear map if $e(g_1, g_2)$ generates G_t and e is efficiently computable.
Implication: Since $\widehat{e}$ is bilinear , the map $\widehat{e}$ is also Symmetric.
Proof: Being symmetric means that for any $Q, W \in G_1$, the equality $\widehat{e}(Q, W) = \widehat{e}(W, Q)$ holds. Both $Q, W \in G_1$ can be represented as using some generator P and some $a, b \in Z_q^*$: Let $Q = aP$ and $W = bP$. Then we have $\widehat{e}(Q, W) = \widehat{e}(aP, bP) = \widehat{e}(P, P)ab = \widehat{e}(bP, aP) = \widehat{e}(W, Q)$.

What groups to use? Typically, G_1 is a subgroup of the group of points on an elliptic curve over a finite field, i.e. $E(F_t)$. G_2 is a subgroup of the multiplicative group of related finite field. The Map $\widehat{e}$ is derived by modifying Weil pairing [29] or Tate pairing [30] on an elliptic curve over F_t. The Computational complexity of the Tate pairing is less than that of the Weil Pairing. The Weil and Tate pairing need to be modified because the pairings may always output 1 ($\in G_t$).

Chapter 4

A Modified ID Based Generalized Signcryption Scheme(MIDGSC)

In this chapter, we have proposed an identity based signcryption scheme based on Barellto et al. identity based signcryption scheme [31] and it is a modified form of Yu et al. Scheme [24].

4.1 Framework of the Scheme

The Algorithm for the Modified Identity Based Signcryption scheme MIDGSC = (Setup, KeyGenration, GSC, GUSC) consists of four algorithms which are:

- **Setup** (1^k)**:** This is a randomized algorithm run by **PKG**. Given a security parameter k, this algorithm generates the system parameters *params* and master secret key s and master public key *mpk*.
- **KeyGenration** (mpk, msk, ID)**:** On input ID, PKG uses it to compute a pair of corresponding public/private keys (S_U, Q_U).
- **GSC:** To send a message m from Sender S to the Receiver R, this algorithm takes input (S_S, ID_R, m) and outputs signcrypted text $\sigma = MIDGSC(S_S, ID_R, m)$.

  - When $ID_S \neq ID_\phi, ID_R \neq ID_\phi, \sigma \leftarrow GSC(S_S, Q_R, m) = SC(S_S, Q_R, m)$
  - When $ID_S \neq ID_\phi, ID_R = ID_\phi, \sigma \leftarrow GSC(S_S, Q_R, m) = Sign(S_S, m)$
  - When $ID_S = ID_\phi, ID_R \neq ID_\phi, \sigma \leftarrow GSC(S_S, Q_R, m) = Encrypt(Q_R, m)$

- **GUSC:** This algorithm takes input (ID_S, S_R, σ) and outputs m if σ is a valid Generalized Signcryption done by Sender S for Receiver R, otherwise output false ($\perp$) if is not valid.
  - When $ID_S \neq ID_\phi, ID_R \neq ID_\phi, m \leftarrow GUSC(Q_S, S_R, \delta) = USC(Q_S, S_R, \delta)$
  - When $ID_S \neq ID_\phi, ID_R = ID_\phi, (T, \perp) \leftarrow GUSC(Q_S, S_R, \delta) = Verify(S_S, \delta)$
  - When $ID_S = ID_\phi, ID_R \neq ID_\phi, m \leftarrow GUSC(Q_S, S_R, \delta) = Decrypt(Q_R, \delta)$

The absence of specific sender or receiver are denoted by ID_ϕ, ID_ϕ instead of ID_S, ID_R. When there is no specific sender(ID_ϕ) we only encrypt the message m using MIDGSC, when information about sender is not needed MIDGSC becomes signature scheme and when both are there it will work as Signcryption scheme.

4.2 Description of the Scheme

Setup: Given a security parameter 1^k, the PKG chooses two groups G_1 and G_2 of prime order p, a random generator P of G_1 and a bilinear map $e : G_1 \times G_1 \rightarrow G_2$, three cryptographic hash functions as:

- $H_0 : \{0,1\}^* \rightarrow Z_p^*$
- $H_1 : G_2 \rightarrow Z_p^*$
- $H_2 : \{0,1\}* \rightarrow Z_p^*$

Where,n denotes the number of bits to represent a message, **PKG** chooses a random $msk \in Z_p^*$ as master secret key and set $mpk = msk \times P$. A special function f is defined as $f(ID) = 0$ if $ID = ID_\phi$, otherwise $f(ID) = 1$.(Assumptions $H_1(1) = 1, H_0(ID_\phi) = 0)$). Also it is assumed that $Q_\phi = 0$.
PKG publishes the system parameters as as $< G_1, G_2, p, n, P, mpk, f, H_1, H_2, H_3 >$.
KeyGenration: Given a user with identity ID_U, its public key is $Q_U = H_0(ID_U)$ is a simple transformation of its Identity. The private key is generated by the PKG as $S_U = sQ_U$.
Generalized Signcryption(GSC): If the Sender S with Identity ID_S has to send a message to the Receiver R with identity ID_R , it does as follows

- Computes $f(ID_S)$ and $f(ID_R)$
- Selects r uniformly from Z_p^* and computes
  - $U \leftarrow rP$
  - $W \leftarrow e(mpk, Q_R)^{rf(ID_R)}$
  - $h_1 \leftarrow H_1(W)$
  - $h_2 \leftarrow H_2(U, W, m, Q_S, Q_R, ID_S, ID_R)$
  - $V \leftarrow h_2 P + f(ID_S)h_1 S_S$
  - $X \leftarrow rV$
  - $Q_R \leftarrow H_0(ID_R)$
  - $y \leftarrow m \parallel ID_S \parallel X \oplus h_1 f(ID_R)$
- Return(U,y)

Generalized UnSigncryption(GUSC): After Receiving(U,y) the receiver computes

- $f(ID_R)$
- $W \leftarrow e(U, S_R)^{f(ID_R)}$
- $h_1 \leftarrow H_1(W)$

- $m \parallel ID_S \parallel X \leftarrow h_1 f(ID_R) \oplus y$
- $h_2 \leftarrow H_2(U, W, m, Q_S, Q_R, ID_S, ID_R)$

Checks if $e(P, X) \neq e(U, P)^h_2 e(r \cdot mpk, Q_S)^{h_1 f(ID_S)}$ return $\perp$, else return m.

Consistency:

- $W = e(mpk, Q_R)^{rf(ID_R)} = e(sP, Q_R)^{rf(ID_R)}$
  $=e(P, sQ_R)^{rf(ID_R)} = e(rP, S_R)^{f(ID_R)}$
  $=e(U, S_R)^{f(ID_R)}$
- $e(P, X) = e(P, rV) = e(rP, V)$
  $=e(rP, h_2 P + f(ID_S)h_1 S_S)$
  $=e(rP, h_2 P)e(rP, f(ID_S)h_1 S_S)$
  $=e(U, P)^{h_2} e(rP, S_S)^{f(ID_S)h_1}$
  $= e(U, P)^{h_2} e(rP, sQ_S)^{f(ID_S)h_1}$
  $=e(U, P)^{h_2} e(rsP, Q_S)^{f(ID_S)h_1}$
  $=e(U, P)^{h_2} e(rmpk, Q_S)^{f(ID_S)h_1}$

4.2.1 Correctness:

The three modes of the scheme are to be considered

- **Signcryption Mode:** In this case $ID_S \neq ID_\phi, ID_R \neq ID_\phi$, so $f(ID_S) = f(ID_R) = 1$, and the scheme will act as Signcryption scheme. It can be verified that:
  - $W = e(mpk, Q_R)^r = e(sP, Q_R)^r$
    $=e(P, sQ_R)^r = e(rP, S_R)$ $=e(U, S_R)$
  - $e(P, X) = e(P, rV) = e(rP, V)$
    $=e(rP, h_2 P + h_1 S_S)$
    $=e(rP, h_2 P)e(rP, h_1 S_S)$
    $=e(U, P)^{h_2} e(rP, S_S)^{h_1}$
    $= e(U, P)^{h_2} e(rP, sQ_S)^{h_1}$

$=e(U,P)^{h_2}e(rsP,Q_S)^{h_1}$

$=e(U,P)^{h_2}e(rmpk,Q_S)^{h_1}$

- **Signature Mode:** In this mode $ID_S \neq ID_\phi, ID_R = ID_\phi$, that is $f(ID_S)=1, f(ID_R)=0$. The Scheme will work as a Signature Scheme.

  - **Sign:**

    Choose random $r \in Z_p^*$

    Compute :

    * $U \leftarrow rP$
    * $W \leftarrow e(mpk,Q_R)^0 = 1$
    * $h_1 \leftarrow H_1(1) = 1$
    * $h_2 \leftarrow H_2(U,1,m,Q_S,Q_\phi,ID_S,ID_\phi)$
    * $V \leftarrow h_2P + f(ID_S)h_1S_S = h_2P + S_S$
    * $X \leftarrow rV$
    * $Q_R \leftarrow H_0(\phi) = 0$
    * $y \leftarrow m \parallel ID_S \parallel X \oplus 0 = m \parallel ID_S \parallel X \oplus 0$

    Return(U,y)

  - **Verify:**

    After Receiving (U,y) the receiver computes

    * $W \leftarrow 1$
    * $h_1 \leftarrow H_1(1) = 1$
    * $m \parallel ID_S \parallel X = y \oplus 0$
    * $h_2 \leftarrow H_2(U,1,m,Q_S,Q_\phi,ID_S,ID_\phi)$

    Check if$e(P,X) \neq e(U,P)^{h_2}e(rmpk,Q_S)^{h_1 f(ID_S)}$ return $\perp$

- **Encryption Mode:**In this mode $ID_S = ID_\phi, ID_R \neq ID_\phi$, that is $f(ID_S)=1, f(ID_R)=0$. The Scheme will work as an Encryption Scheme.

- **Encrypt:**
  - $U \leftarrow rP$, $W \leftarrow e(mpk, Q_R)^r$
  - $h_1 \leftarrow H_1(W)$
  - $h_2 \leftarrow H_2(U, W, m, Q_\phi, Q_R, ID_\phi, ID_R)$
  - $V \leftarrow h_2 P + 0 = h_2 P$
  - $X \leftarrow rV$
  - $Q_R \leftarrow H_0(ID_R)$
  - $y \leftarrow m \parallel 0 \parallel X \oplus h_1$

  Return (U, y)
- **Decrypt:**Computes $f(ID_R)$ and also Computes
  - $W \leftarrow e(U, S_R)$
  - $h_1 \leftarrow H_1(W)$
  - $m \parallel 0 \parallel X = h_1 \oplus y$
  - $h_2 \leftarrow H_2(U, W, m, Q_\phi, Q_R, ID_\phi, ID_R)$

  Checks if $e(P, X) \neq e(U, P)^{h_2}$ return $\perp$, else return m.

4.3 Efficiency analysis

The basic purpose of generalized signcryption is to reduce implementation complexity. As per need in different application environments, generalized signcryption can fulfill the function of signature, encryption or signcryption respectively. However, the computational and communication cost may increase compared with the normal signcryption schemes. The proposed scheme significantly reduces the extra computations and has comparable efficiency as compared to the existing efficient identity based signcryption schemes [12, 32, 33]. In Table 4.1 we compare the computational complexity of our scheme with several other efficient existing signcryption schemes. Moreover, we compare our efficiency with other existing identity based generalized signcryption schemes [22, 24]. Our scheme gives better performance as compared to IDGSC [22],and gives comparable

efficiency as compared to NIDGSC [24]. Also,the proposed scheme uses less number of schemes as compared to other ID based generalized signcryption.

Table 4.1: Efficiency Comparison with other Signcryption schemes

Schemes	Signcryption			UnSigncryption		
	M	E	P	M	E	P
Malone Lee's	3	0	0(+1)	0	1	3(+1)
Libert Quisquater's	2	2	0(+2)	0	2	3(+2)
X Boyen's	3	1	0(+1)	2	0	3(+1)
Chow et al.'s	2	0	0(+2)	1	0	4
Proposed Scheme	3	1	0(+1)	1	2	2(+2)

M: number of point multiplications in G_1; E: number of exponentiation in G_2; P:number of pairing computations; (+): pre-computation of pairing.

The Table 4.1 shows that the proposed scheme has comparable efficiency as compared to other existing signcryption schemes. Almost with same computational cost, the proposed can work as a signcryption scheme when both confidentiality and authentication are needed and as an encryption scheme or a signature scheme when anyone them is needed.

Table 4.2: Efficiency Comparison with other IDGSC schemes

Schemes	Generalized Signcryption				Generalized UnSigncryption			
	M	E	P	H	M	E	P	H
IDGSC	5	0	0(+1)	3	1	0	3(+1)	3
NIDGSC	3	1	0(+1)	4	0	2	2(+2)	3
Proposed Scheme	3	1	0(+1)	3	1	2	2(+2)	2

M: number of point multiplications in G_1; E: number of exponentiation in G_2; P:number of pairing computations; H:number of hash function; (+): pre-computation of pairing.

The Table 4.2 shows that the proposed scheme has better efficiency as compared to the IDGSC,and has comparable efficiency with respect NIDGSC. Overall as compared to all the existing scheme the proposed scheme uses less no of Hashing and hence it has got better efficiency than other schemes.

Chapter 5

An improved Certificateless Generalized Signcryption scheme

In this chapter, we have proposed an certificateless generalized signcryption scheme based on Barbosa et al. certificateless signcryption scheme.

5.1 Framework of improved CLGSC

This scheme consists of six algorithms. First four of which are used for key management operations.

1. **Setup**(1^k)**:**This is a global setup algorithm, which takes input the security parameter 1^k and returns the **KGC**'s secret key **msk** and global parameters params including a master public key **mpk**. This algorithm is executed by the **KGC**, which publishes *params*.

2. **Extract-partial-private-key** $(ID_U, msk, params)$**:** An algorithm which takes input *msk*, *params* and a user identity $ID_U \in \{0,1\}^*$ and returns a partial private key D_U. This algorithm is run by **KGC**, after verifying the users identity.

3. **Generate-User-Keys** $(ID_U, params)$**:** An algorithm which takes input as an identity and the public parameters and outputs a secret value x and a

public key PK. This algorithm is run by a user to obtain a public key and a secret value which will be used for constructing full private key. The public key is published without certification.

4. **Set-Private-Key ($D_U, x, params$):** A deterministic algorithm which takes as input a partial secret key D_U and a secret value x and returns the full private key S_U. This algorithm is run by a user to construct a full private key.

5. **CLGSC (m, S_S, ID_R):** This algorithm has three scenarios: signcryption mode, signature only mode and encryption only mode.

   - **Signcryption Mode:** If sender S transmits wants to transmit a message m to receiver B such that both confidentially and authentication need to be maintained then the input is (m, S_S, ID_R), and output is $\sigma = CLGSC(m, S_S, ID_R) = Singcrypt(m, S_S, ID_R)$.
   - **Signature only Mode:** If sender S wants to send message m without definite receiver, the input is (m, S_S, ID_ϕ), where ID_ϕ means receiver is null, the output is$\sigma = CLGSC(m, S_S, ID_\phi) = sign(m, S_S)$.
   - **Encryption only Mode:** If someone wants to send a message m to a definite receiver R confidentially, the input is (m, S_ϕ, ID_R), where S_ϕ means the receiver is null, the output is$\sigma = encrypt(m, ID_R)$.

6. **CLGDSC (σ, ID_S, ID_R):** After receiving σ, if it is valid, the receiver R designcrypts (or decrypts) the ciphertext and returns the message m and (or) the signature on m by S, otherwise return ($\perp$) means false.

5.2 Description of the Proposed CLGSC scheme

In this section we proposed a new CLGSC scheme based on the Certificateless Signcryption scheme proposed in [Barbosa et. al] scheme.

- **Setup** (1^k)**:** Given a security parameter k, the **KGC** chooses two groups G_1, G_2 of prime order p, a random generator P of G_1, a bilinear map $e : G_1 \times G_1 \to G_2$, four cryptographic hash functions as:

  - $H_1 : \{0,1\}^* \to G_1$
  - $H_2 : \{0,1\}^* \to \{0,1\}^n$
  - $H_3 : \{0,1\}^* \to G_1$

  Where, n denotes the number of bits to represent a message. A special function f is defined as $f(ID) = 0$, if $ID = ID_\phi$ otherwise $f(ID) = 1$. $ID_\phi, PK_\phi$and $S_\phi$are parsed as strings of zero.**KGC** chooses a random $msk \in Z_p^*$ as master secret key and set $mpk = msk \times P$. KGC publishes the system parameters as as$< G_1, G_2, p, n, P, mpk, f, H_1, H_2, H_3 >$.

- **Extract-Partial-Private-Key:** Given a user with identity ID_U, the partial private key is computed by KGC as $D_U = mskQ_U = mskH_1(ID_U)$.

- **Generate-User-Keys:** Given D_U, the user with identity ID_U chooses a random $x_U \in Z_p^*$ and sets its public key $PK_U = x_U P$ and Private Key $S_U =< x_U, D_U >$.

- **CLGSC** $(m, ID_S, ID_R, S_S, PK_S, PK_R, mpk)$

  1. Computes $f(ID_S)$ and $f(ID_R)$,Selects r uniformly from Z_p^*
  2. Compute
     - $U \leftarrow rP$, $T \leftarrow e(mpk, Q_R)^r f(ID_R)$
     - $h \leftarrow H_2(U, T, rPK_R, ID_S, ID_R, PK_S, PK_R) f(ID_R)$
     - $V \leftarrow m \oplus h$
     - $H \leftarrow H_3(U, V, ID_S, ID_R, PK_S, PK_R)$
     - $W \leftarrow f(ID_S)[D_S + x_S H] + rH$
  3. Return $\sigma \leftarrow (U, V, W)$

- **CLGDSC** $(\sigma, ID_S, ID_R, S_S, PK_S, PK_R, mpk)$ After Receiving σ from Sender S, the receiver R parses σ as U,V,W and

  1. Computes $f(ID_R), f(ID_S)$
  2. Computes $H \leftarrow H_3(U, V, ID_S, ID_R, PK_S, PK_R)^{f(ID_R)}$
  3. Check if $e(P, W) \neq e(mpk, Q_S)^{f(ID_S)} e(U + PK_S, H)$ return $\perp$ else computes
     - $T \leftarrow (U, D_R)$, parse S_R as (x_R, D_R)
     - $h \leftarrow H_2(U, T, x_R U, ID_S, ID_R, PK_S, PK_R)$
     - $m \leftarrow V \oplus h$
  4. Return m

- **Consistency:**

  - $T = e(U, D_R) = e(rP, mskQ_R) = e(rP, Q_R)^{msk} = e(r \cdot msk \cdot P, Q_R) = e(r \cdot mpk, Q_R) = e(mpk, Q_R)^r$
  - $e(P, W) = e(P, D_S + x_S \cdot H + rH) = e(P, mskH_1(ID_S)) e((r + x_S)P, H) = e(mskP, H_1(ID_S)) e(U + PK_S, H) = e(mpk, Q_S) e(U + PK_S, H)$

5.2.1 Adaptability and Correctness:

CLGSC is an adaptive scheme and can work as Signcryption scheme, Signature Scheme and Encryption Scheme depending on the need whether Confidentiality and Authentication are needed simultaneously or individually, without any other additional operation.

- **Signcryption Mode:** When $ID_S \neq ID_\phi, ID_R \neq ID_\phi then f(ID_S) = 1, f(ID_R) = 1$, the algorithm runs in signcryption mode.
- **Signature only Mode:** When $ID_S \neq ID_\phi, ID_R = ID_\phi then f(ID_S) = 1, f(ID_R) = 0$, the algorithm runs in signature mode. For this the CLGSC and CLGDSC becomes

- **CL-Signature** $(m, ID_S, ID_\phi, S_S, PK_S, PK_\phi, mpk)$
  - $U = rP, T = 1, h = 0, V = m \oplus 0 = m$
  - $H = H_3(U, V, ID_S, ID_\phi, PK_S, PK_\phi)$
  - $W = f(ID_S)[D_S] + x_S H] + rH$

  Return $\sigma \leftarrow (U, m, W)$, where (U, W) is the signature on m
- **CL-Verify**$(\sigma, ID_S, ID_\phi, S_R, PK_S, ID_\phi, mpk)$

  On receiving U,m,W the receiver computes
  - $H = H_3(U, m, ID_S, ID_\phi, PK_S, ID_\phi)$
  - Verify if $e(P, W) \neq e(mpk, Q_S)^{f(ID_S)} e(U + PK_S, H)$ return $\perp$ else accept the message.

- **Encryption only Mode:**When $ID_S = ID_\phi, ID_R \neq ID_\phi$ then $f(ID_S) = 0, f(ID_R) = 1$, the algorithm runs in encryption mode. CLGSC and CLGDSC becomes:
  - **CL-Encrypt**$(m, ID_\phi, ID_R, S_\phi, PK_\phi, PK_R, mpk)$
    - $U = rP$, $T \leftarrow e(mpk, Q_R)^r$
    - $h = H_2(U, T, rPK_R, ID_\phi, ID_R, ID_\phi, PK_R)$
    - $V \leftarrow m \oplus h$
    - $H \leftarrow H_3(U, V, ID_\phi, ID_R, PK_\phi, PK_R)$
    - $W \leftarrow 0[D_S + x_S H] + rH{=}rH$

    Return $\sigma \leftarrow (U, V, W)$.
  - **CL-Decryption**$(\sigma, ID_\phi, ID_R, S_\phi, PK_\phi, mpk)$ on receiving U,m,W the receiver computes
    - $H = H_3(U, V, \phi, ID_R, \phi, PK_R)$
    - if $e(P, W) \neq 1e(U + 0, H) \neq e(U, H)$ return $\perp$ else computes
      - $T = e(U, D_R)$, parse S_R as (x_R, D_R)
      - $h \leftarrow H_2(U, T, x_R U, ID_\phi, ID_R, PK_\phi, PK_R)$
      - $m \leftarrow V \oplus h$

    Return m

5.3 Efficiency Analysis

Computation time and ciphertext size are two important parameters affecting the efficiency of a cryptographic scheme. We present a comparison of our scheme with other existing CLGSC schemes with respect to these parameters.

Table 5.1: Efficiency Comparison with Certificateless Signcryption Scheme

Schemes	Ciphertext Size	Signcryption				DeSigncryption			
		E	M	P	H	E	M	P	H
Barbosa et al. Scheme	$2\|G_1\| + \|m\|$	1	4	0(+1)	3	0	1	4(+1)	3
Proposed Scheme	$2\|G_1\| + \|m\|$	1	4	0(+1)	3	0	1	4(+1)	3

M: number of point multiplications in G_1; E: number of exponentiation in G_2; P:number of pairing computations; H:number of hash function; (+): pre-computation of pairing; $|G_1|$:Size of an element in G_1 ; $|G_2|$:Size of an element in G_2; $|m|$:length of message m; $|ID|$:length of identity; $|p|$:Size of an element inZ_p^*.

The Table 5.1 shows that Barbosa et al.'s signcryption scheme [34] has the same ciphertext size and efficiency as our scheme. That means both the schemes have the same Computation Complexity and Communication Complexity. But in terms of implementation complexity our scheme is a better than first one because, Barbosa et al.'s Certificateless Signcryption [34] Scheme cannot work as signature only or encryption only mode, but our scheme can adaptively work as a signcryption scheme when both confidentiality and authentication are needed and as an encryption scheme or a signature scheme when anyone them is needed.

M: number of point multiplications in G_1; E: number of exponentiation in G_2; P:number of pairing computations; H:number of hash function; (+): pre-computation of pairing; $|G_1|$:Size of an element in G_1 ; $|G_2|$:Size of an element in G_2; $|m|$:length of message m; $|ID|$:length of identity; $|p|$:Size of an element inZ_p^*.

Table 5.2: Efficiency Comparison with other CLGSC Schemes

Schemes	Ciphertext Size	GSC				GDSC			
		E	M	P	H	E	M	P	H
Ji et al. [26]	$2\|G_1\| + \|m\| + \|ID\| + \|G_2\| + \|P\|$	3	2	0	4	1	1	2	4
Kushwah et al. [27]	$2\|G_1\| + \|m\| + \|ID\| + \|G_2\|$	2	3	0	3	1	3	2	3
Zhou et al. [28]	$2\|G_1\| + \|m\|$	1	4	0(+1)	3	0	1	4(+1)	3
Proposed Scheme	$2\|G_1\| + \|m\|$	1	4	0(+1)	2	0	1	4(+1)	2

The Table 5.2 shows that the proposed scheme has smaller text size as compared to first two schemes but has same size as third scheme. But as compared to all the existing scheme our scheme uses less no of Hashing and hence it has got better efficiency than other schemes.

Chapter 6

Conclusion

6.1 Conclusion and Future Work

Generalized Signcryption is a multi functional single subroutine which can adaptively work as an encryption scheme or a signcryption scheme or a signcryption scheme. It is Suitable for resource constrained environment like: Adhoc Networks, WSNs, Mobile Computing and Ubiquitous Computing,Embedded Systems. According to the comparison to other schemes, the proposed schemes are efficient. Due to the Computation of the pairing being still time consuming the schemes can be further improved by reducing no of pairing operations at the same time maintaining the efficiency. Finally, the proposed scheme can also be extended for multiuser environment, broadcast communication.

Bibliography

[1] Behrouz A Forouzan. *Cryptography & Network Security*. McGraw-Hill, Inc., 2007.

[2] Alexander W Dent. *Practical signcryption*. Springer, 2010.

[3] Yuliang Zheng. Digital signcryption or how to achieve cost (signature & encryption) cost (signature)+ cost (encryption). In *Advances in CryptologyCRYPTO'97*, pages 165–179. Springer, 1997.

[4] Yuliang Zheng and Hideki Imai. How to construct efficient signcryption schemes on elliptic curves. *Information Processing Letters*, 68(5):227–233, 1998.

[5] Yiliang Han. Generalization of signcryption for resources-constrained environments. *Wireless Communications and Mobile Computing*, 7(7):919–931, 2007.

[6] Ren-Junn Hwang, Chih-Hua Lai, and Feng-Fu Su. An efficient signcryption scheme with forward secrecy based on elliptic curve. *Applied Mathematics and computation*, 167(2):870–881, 2005.

[7] Yiliang Han, Xiaoyuan Yang, Ping Wei, Yuming Wang, and Yupu Hu. Ecgsc: elliptic curve based generalized signcryption. In *Ubiquitous Intelligence and Computing*, pages 956–965. Springer, 2006.

[8] Yliang Han and Xiaolin Gui. Bpgsc: Bilinear paring based genearlized signcryption scheme. In *Grid and Cooperative Computing, 2009. GCC'09. Eighth International Conference on*, pages 76–82. IEEE, 2009.

[9] Joonsang Baek, Ron Steinfeld, and Yuliang Zheng. Formal proofs for the security of signcryption. In *Public Key Cryptography*, pages 80–98. Springer, 2002.

[10] John Malone-Lee and Wenbo Mao. Two birds one stone: signcryption using rsa. In *Topics in CryptologyCT-RSA 2003*, pages 211–226. Springer, 2003.

[11] Yuliang Zheng. Identification, signature and signcryption using high order residues modulo an rsa composite. In *Public Key Cryptography*, pages 48–63. Springer, 2001.

[12] Benoıt Libert and Jean-Jacques Quisquater. A new identity based signcryption scheme from pairings. In *Information Theory Workshop, 2003. Proceedings. 2003 IEEE*, pages 155–158. IEEE, 2003.

[13] Xavier Boyen. Multipurpose identity-based signcryption. In *Advances in Cryptology-CRYPTO 2003*, pages 383–399. Springer, 2003.

[14] DongJin Kwak, SangJae Moon, Guilin Wang, and Rorbert H Deng. A secure extension of the kwak–moon group signcryption scheme. *computers & security*, 25(6):435–444, 2006.

[15] Yiliang Han and Xiaoyuan Yang. Ecgsc: Elliptic curve based generalized signcryption scheme. *IACR Cryptology ePrint Archive*, 2006:126, 2006.

[16] Jindan Zhang and Xu an Wang. Formal security proof for generalized signcryption. In *E-Business and Information System Security, 2009. EBISS'09. International Conference on*, pages 1–5. IEEE, 2009.

[17] Xiaoyuan Yang, Maotang Li, Lixian Wei, and Yiliang Han. New ecdsa-verifiable multi-receiver generalization signcryption. In *High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on*, pages 1042–1047. IEEE, 2008.

[18] HF Ji, WB Han, and Long Zhao. Identity-based generalized signcryption in standard model. *Appl. Res. Comput*, 27(10):3851–3854, 2010.

[19] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in cryptology*, pages 47–53. Springer, 1985.

[20] Sattam S Al-Riyami and Kenneth G Paterson. Certificateless public key cryptography. In *Advances in Cryptology-ASIACRYPT 2003*, pages 452–473. Springer, 2003.

[21] Zhang Chuanrong, Chi Long, and Zhang Yuqing. Secure and efficient generalized signcryption scheme based on a short ecdsa. In *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2010 Sixth International Conference on*, pages 466–469. IEEE, 2010.

[22] Sunder Lal and Prashant Kushwah. Id based generalized signcryption, 2008.

[23] Sunder Lal and Prashant Kushwah. Generalization of barreto et al id based signcryption scheme.

[24] Gang Yu, Xiaoxiao Ma, Yong Shen, and Wenbao Han. Provable secure identity based generalized signcryption scheme. *Theoretical Computer Science*, 411(40):3614–3624, 2010.

[25] Prashant Kushwah and Sunder Lal. An efficient identity based generalized signcryption scheme. *Theoretical Computer Science*, 412(45):6382–6389, 2011.

[26] Huifang Ji, Wenbao Han, and Long Zhao. Certificateless generalized signcryption. *Physics Procedia*, 33:962–967, 2012.

[27] Prashant Kushwah and Sunder Lal. Efficient generalized signcryption schemes, 2010.

[28] Caixue Zhou, Wan Zhou, and Xiwei Dong. Provable certificateless generalized signcryption scheme. *Designs, Codes and Cryptography*, pages 1–16, 2012.

[29] Victor S Miller. The weil pairing, and its efficient calculation. *Journal of Cryptology*, 17(4):235–261, 2004.

[30] Michael Scott. Computing the tate pairing. In *Topics in Cryptology–CT-RSA 2005*, pages 293–304. Springer, 2005.

[31] Paulo SLM Barreto, Benoît Libert, Noel McCullagh, and Jean-Jacques Quisquater. Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. In *Advances in Cryptology-ASIACRYPT 2005*, pages 515–532. Springer, 2005.

[32] John Malone-Lee. Identity-based signcryption, 2002.

[33] Sherman SM Chow, Siu-Ming Yiu, Lucas CK Hui, and KP Chow. Efficient forward and provably secure id-based signcryption scheme with public verifiability and public ciphertext authenticity. In *Information Security and Cryptology-ICISC 2003*, pages 352–369. Springer, 2004.

[34] Manuel Barbosa and Pooya Farshim. Certificateless signcryption. In *Proceedings of the 2008 ACM symposium on Information, computer and communications security*, pages 369–372. ACM, 2008.