Implementation of Elgamal Encryption in Electronic Voting

By

Shahid Mehmood

CIIT/FA16-RMT-008/LHR

MS

In

Mathematics

COMSATS University Islamabad,

Lahore Campus-Pakistan

Spring, 2018

COMSATS University Islamabad, Lahore Campus

Implementation of Elgamal Encryption in Electronic Voting

A thesis Presented to

COMSATS University Islamabad,

Lahore Campus

In partial fulfillment

of the requirement for the degree of

MS (Mathematics)

By

Shahid Mehmood

CIIT/FA16-RMT-008/LHR

Spring, 2018

Implementation of Elgamal Encryption in Electronic Voting

A Thesis is submitted to the Department of Mathematics as partial fulfillment of the requirement for the award of Degree of MS Mathematics.

Name	Registration Number
SHAHID MEHMOOD	CIIT/FA16-RMT-008/LHR

Supervisor

Dr. Tariq Javed Zia

Assistant Professor,
Department of Mathematics
COMSATS University Islamabad,
Lahore Campus

July, 2018

Final Approval

This thesis titled

Implementation of Elgamal Encryption in Electronic Voting

By

Shahid Mehmood

CIIT/FA16-RMT-008/LHR

Has been approved

For the COMSATS University Islamabad (CUI), Lahore Campus.

External Examiner:_____

Prof. Dr. Akhlaq Ahmad Bhatti

National University of Computer and Emerging Sciences, Lahore Campus

Supervisor:_____

Dr. Tariq Javed Zia

Department of Mathematics, (CUI) Lahore Campus

HoD:_____

Dr. Sarfraz Ahmad

Department of Mathematics, (CUI) Lahore Campus

Declaration

I **Shahid Mehmood** registration number **CIIT/FA16-RMT-008/LHR,** hereby declare that I have produced the work presented in this thesis, during the scheduled period of study. I also declare that I have not taken any material from any source except referred to wherever do that amount of plagiarism is within acceptable range. If a violation HEC rules has occurred in this thesis, I shall be liable to punishable action under the plagiarism rules of the HEC.

Date:_____

_____

Shahid Mehmood

CIIT/FA16-RMT-008/LHR

Certificate

It is certified that Shahid Mehmood (CIIT/FA16-RMT-008/LHR) has carried out all the work related to this thesis under my supervision at the Department of Mathematics, COMSATS University Islamabad, Lahore Campus, and the work fulfills the requirement for award of MS degree.

Date:_____

Supervisor:

_____

Dr. Tariq Javed Zia
Assistant Professor
Department of Mathematics
COMSATS University
Islamabad,
Lahore Campus

Head of Department:

_____

Dr. Safraz Ahmad
Associate Professor
Department of Mathematics
COMSATS University Islamabad,
Lahore Campus

DEDICATION

To,

Every challenging work need self efforts as well guidance elders specially those who were very close to our heart. I dedicate this thesis to my Father Azeem Khan and my loving mother. Whose affection, love, encouragement and prayers of day and night make me able to get such success and honor. I am also thankful to my class mate M. Sarfraz Khan, and Farwa Afridi to help me during my course work.

ACKNOWLEDGEMENTS

In the name of Allah, the Most Gracious and the Most Merciful
The Prophet (S.A.W) said:
"Seeking knowledge is obligatory for every Muslim"

First, I recognize the creator and finisher of my faith and in memorial of my life, I say thank ALLAH Almighty and Prophet Muhammad (S.A.W) for everything. I am grateful to Almighty ALLAH, without Whose endowments it was difficult to finish this proposal.

After this, I wish to thank my supervisor Dr. Tariq Javed Zia who did not discriminate about day or night to help me in my research work. It has been a pleasure to have him as my supervisor. He is always ready to bestow his significant apprehension when needed. I am also thankful for his convenient and vigilant analyze to revamp the aspect of the thesis. His direction, kind supervision and productive recommendations empowered me to finish this work. I am appreciative to Dr. Sarfraz Ahmad, Head of Department of Mathematics, for his direction amid course work.

I am appreciative to my family for their steady care, tolerance and love. Finally, I am appreciative my elder brother Waseem Akram for their guidance and commitment. Disregarding them this work could never have appeared (actually).

At the end I would acknowledge the pleasant moments shared with my fellows specially Muhammad Sarfraz, Ayesha Liaquat, Sawaira Afridi, Sobia Khanum and all fellows.

Shahid Mehmood

CIIT/FA16-RMT-008/LHR

Abstract

In this dissertation, in general we have studied the implementation of a finite groups in cryptography, especially theoretical based being have mathematical background. In particular the Elgamal encryption scheme based on finite groups has been used being more robust and efficient. It operates on a cyclic group for which the decisional Diffi-Hellman (DDH) assumption holds. This group is the subgroup of Quadratic residue $G_q < Z_p^*$ of prime order q, where p = 2q + 1 is a safe prime. It is obvious from the definition of Elgamal encryption, which is public key encryption scheme, that the keys generated by the multiple parties (of course finite in numbers) can be combined to form a single (common) public key. The corresponding private keys can be regarded as key shares and, are used in collaborative manner. Consequently we have studied the implementation of modulo groups (Elgamal Algorithm) in particular, in electronic voting by presenting the construction of receipt free protocol from a given primary voting scheme. After applying this construction to the voting protocol we obtain an efficient receipt free 1-out-of-L voting protocol based on homomorphic encryption.

TABLE OF CONTENTS

.

LIST OF FIGURES

Chapter 1

Introduction

1.1 Overview

The properties of suggested eletronic voting protcols, used currently, based on Public Key Cryptography (PKC), which is the most suitable for key agreement protocols and authentication mechanisms. In PKC, the pair of encryption key e which is known to public and the decryption d which is the secret or private key, are used. The voters use the key (the public key) e to cast the vote and use key (the secret key) d to verify his/her vote being casted to the intended candidate.

To the bind public key to entities (key available for voter) and, enabling their verification, the implement Certification Authority (CA) is necessary for certification while using PKC. The Identity-Based Encryption (IBE) is an alternative way is (to use) from which

"we can have all the benefits offered by PKC, without neither the need of certificates nor all the core components of a Public Key Infrastructure (PKI)".

PKC, firstly, was proposed by "the mathematician Rivest, Shamir and Adleman, called the RSA cryptosystem". RSA is the number theoretic based encryption in which we take $n = pq$ where p, q are kept secret and n is the public key which is known publically and is being chosen the product large prime numbers. The ciphered text $c = x^e \ mod \ n$ and plaintext or original message x is obtained $x = c^d \ mod \ n$. The keys e and d are generated by using the little Fermat's Theorem as follows

• Compute $n = pq$ by taking two random large primes equal in magnitude and different in binary representation to make harder to guess.

• Compute $m = lcm(p-1, q-1)$ kept it secret.

• Choose e such that $gcd(e, m) = 1 \ , 1 < e < m$ Determine integer d such that $ed \equiv 1 \ mod \ m$ or $d = e^{(-1)} \ mod \ m$ the modular multiplicative inverse.

Currently the ElGamal encryption scheme is used being more robust and efficient. It works on an cyclic group for which the decisional Diffie-Hellman

(DDH) assumption holds. This group is the subgroup of the quadratic residues $G_q < \mathbb{Z}_p$ of prime order q. Where $p = 2q + 1$ is safe prime.

The public parameters of an ElGamal encryption scheme are thus p, q and generator α excluding 1. The key pair secret key b and public key β contain a random decryption key b and $\beta \equiv \alpha^b \, mod \, p$ as public key. The detail is as follows

Describe the cyclic group G of order q and generator g. Choose random number b from $\{1, 2, ..., p-2\}$ a secret key Compute $\beta \equiv g^r \, mod \, p$, a public key along with p, α, β. Encrypt the plaintext x under public key (p, α, β) Choose random a from $\{2, ..., p-2\}$ to compute $K_E \equiv \alpha^a \, mod \, p$ Compute $k \equiv \beta^a \, mod \, p$, and $y \equiv xk \, mod \, p$ and map secret message x on to y Calculate $k \equiv K_E \, mod \, p$ and then calculate $x \equiv yk^{-1} \, mmod \, p$ which gives back the original plaintext x.

We further study the modular group's implementation in cryptography in general and e-voting in particular.

In this thesis comprises of three chapters. In chapter one we have given the overview an introduction of our studies. Chapter two deals with the literature relevant to cryptography. Owing to that we have divided this chapter into two main sections. Sections 1 deals with the initial and preliminary number theoretic definitions and theorems while in section 2 we have reproduced the literature over the cryptography and cryptanalysis, being the main branches of information security. While in the last chapter, we will be studying, electronic voting (e-voting), receipt-free e-voting in particular. In electronic voting, the Public Key Cryptography (PKC), along with Certificate Authority (CA) for bind, public keys or Identity Based Entity (IBE) are used.

Our study is focused on IBE which is not required CA. Elgamal scheme which is based on cyclic group, initially, we will be studying the definitions directly used in e-voting then implementation of Elgamal in e-voting.

Chapter 2

Preliminary Mathematical Definitions

In this chapter we have reviewed the literature relevant to cryptography. Owing to that we have divided this chapter into two main sections. Sections 1 deals with the initial and preliminary number theoretic definitions and theorems while in section 2 we have reproduced the literature over the cryptography and cryptanalysis, being the main branches of information security.

2.1 Basic Number Theory

2.1.1 Number System

The most common number systems are decimal, binary, octal, hexadecimal. In above number system decimal number system is most familiar. These systems are classified by their base values. Now octal number system has base value 8, hexadecimal number system has base value 16, binary has base value 2 and eacimal has base value 10.

Sometimes we may required to convert the number systems into any other number system, i-e. We can convert binary to octal, binary to decimal, decimal to binary, binary to hexadecimal, hexadecimal to octal, octal to binary and so on.

Conversion of Decimal-to-binary

Example

Convert $(21)_{10}$ into $(?)_2$.

Division	Quotient	Generated Remainder
$\frac{21}{2}$	10	1
$\frac{10}{2}$	5	0
$\frac{5}{2}$	2	1
$\frac{2}{2}$	1	0
$\frac{1}{2}$	0	1

The required binary number is $(10101)_2$.

Conversion of Decimal-to-octal

Example

Convert $(508)_{10}$ into $(?)_8$.

Division	Quotient	Generated Remainder
$\frac{508}{8}$	63	4
$\frac{63}{8}$	7	7
$\frac{7}{8}$	0	7

The required octal number is $(774)_8$.

Conversion of Decimal-to-hexadecimal

Example

Convert $(357)_{10}$ into $(?)_{16}$.

Division	Quotient	Generated Remainder
$\frac{357}{16}$	22	5
$\frac{22}{16}$	1	6
$\frac{1}{16}$	0	1

The required Hexadecimal number is $(774)_{16}$.

Conversion of Binary-to-decimal

Example

Convert $(1100)_2$ into $(?)_{10}$.

The binary number given is 1 1 1 1 0

Positional weights 3 2 1 0

The decimal number would be:

$1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^1 + 0 \times 2^0 = 8 + 4 + 0 + 0 = (12)_{10}$.

Conversion of Octal-to-decimal

Example: Convert $(2120)_8$ into $(?)_{10}$.

The octal number given is 2 1 2 0 0

Positional weights 4 3 2 1 0

Hence the decimal equivalent number:

$2 \times 8^4 + 1 \times 8^3 + 2 \times 8^2 + 2 \times 8^1 + 0 \times 8^0 = 8192 + 512 + 128 + 16 + 0 = (8848)_{10}$.

Conversion of Hexadecimal-to-decimal

Example: Convert $(31F)_{16}$ into $(?)_{10}$.

The hexadecimal number given is 3 1 F

Positional weights 2 1 0

The decimal equivalent number is:

$3 \times 16^2 + 1 \times 16^1 + 15 \times 16^0 = 768 + 16 + 15 = (799)_{10}$.

2.1.2 Greatest Common Divisor

Let a, b be two integers then d is the G.C.D of a, b if

- $d > 0$.
- $d \mid a$ and $d \mid b$.
- If $c \mid d$ then c is the common divisor.

2.1.3 Euclidean Algorithm

It is a an algorithm in which we compute the (GCD) of two numbers, where (GCD) is the common divisor.

2.1.4 Extended Euclidean Algorithm

Let $(b, c) = d$, where d is the gcd of integeres b and c, where d can be written as $d = ax + by$. So basically extended euclidean algorithm can find those integers x and y.

2.1.5 Fermat's Little Theorem

let $(a, p) = 1$, where p is a prime number and a is a positive integer then $a^{p-1} \equiv 1 \; mod \; p$ or $a^p \equiv a \; mod \; p$

Example

When $p = 5$, $S = \{1, 2, 3, 4\}$. Taking $a = 2$, if we multiply the elements os S by 2 i-e $2S = \{2, 4, 6, 8\}$. This is not the same set of integers as S, but 2S turns into S when we reduce it mod 5:

$2 \equiv 2$, $4 \equiv 4$, $6 \equiv 1$, $8 \equiv 3$, Similarly, $3S = \{3, 6, 9, 12\}$ and, modulo 5, $3 \equiv 3$, $6 \equiv 1$, $9 \equiv 4$, $12 \equiv 2$.

For any $a \not\equiv 0 \; mod \; 5$, the sets {1, 2, 3, 4} and {a, 2a, 3a, 4a} turn into the same list mod 5, so their products are the same modulo 5:

$1 \cdot 2 \cdot 3 \cdot 4 \equiv a \cdot 2a \cdot 3a \cdot 4a \equiv a^4(1 \cdot 2 \cdot 3 \cdot 4) \; mod \; 5$.

Canceling the common factors 1, 2, 3a and 4 from both sides, since they are all nonzero mod 5, we are left with $1 \equiv a^4 \; mod \; 5$

2.1.6 Chinese Remainder Theorem

Let $m_1, ..., m_k$ be positive integers greater than 1 and are relatively prime in pairs. Then the following systen of simaltaneous linear congruences has a unique solution modulo m, where $m = m_1, ..., m_k$

$$x \equiv c_1 \; (mod \; m_2)$$
$$x \equiv c_2 \; (mod \; m_2)$$
$$.$$
$$.$$
$$.$$
$$x \equiv c_k \; (mod \; m_k)$$

Proof let $M_i = \frac{m_1, ..., m_k}{m_i}$, $\quad i = 1, 2, ..., k$ Since $m_1, ..., m_k$ are relatively prime

in pairs, therefore gcd$(M_i, m_i) = 1$. Hence each of the congruences $M_i y_i \equiv 1\ (mod\ m_i)$ has a unique solution for y_i. Now consider the integer:

$y = c_1 M_1 y_1 + c_2 M_2 y_2 + ... + c_k M_k y_k$.

Since $M_j \equiv 0\ (mod\ m_i)$, for $i \neq j$ and $M_i y_i \equiv 1\ (mod\ m_i)$ we observe that $y \equiv c_i\ mod\ m_i$ for $i = 1, 2, ..., k$. Thus the integer $y = c_1 M_1 y_1 + c_2 M_2 y_2 + ... + c_k M_k y_k$ satisfies all the congruences of the system and hence $x \equiv y\ (mod\ m)$ is a solution of it. Now we prove the uniqueness of this solution. Let us suppose that there is another integer z such that $z \equiv c_i \equiv\ (mod\ m_i)$ for $i = 1, 2, ..., k$. Then obviously $z \equiv c_i \equiv y \equiv\ (mod\ m_i)$ for $i = 1, 2, ..., k$. Since the moduli are relatively prime in pairs, therefore $z \equiv c_i \equiv y \equiv\ (mod\ m_i)$. Hence the solution $x \equiv y\ (mod\ m)$ is unique.

Example

We solve the following system of linear congruences:

$$x \equiv 2\ (mod\ 5)$$
$$x \equiv 3\ (mod\ 7)$$
$$x \equiv 5\ (mod\ 11)$$

Here $m = 5(7)(11) = 385$.

$M_1 = 77$, $M_2 = 55$, $M_3 = 35$.

Now the unique solutions of the congruences:

Now we find y_1, y_2, and y_3.

$$77y_1 \equiv 1\ (mod\ 5)$$
$$55y_2 \equiv 1\ (mod\ 7)$$
$$35y_3 \equiv 1\ (mod\ 11)$$

are $y_1 \equiv 2\ (mod\ 5)$, $y_2 \equiv 6\ (mod\ 7)$, $y_3 \equiv 2\ (mod\ 11)$ respectively. So

$y = (2(77)(2)+3(55)(6)+5(35)(2)) = 3298\ x \equiv 3298\ mod\ 231$ or $x \equiv 218\ (mod\ 231)$ is the unique solution of the given system.

2.1.7 Group

A group is a set G, under the binary operation * , if:

- G is closed under this operation.
- G has associative property.
- G contains an identity.
- G contains inverse of every element.

2.1.8 Finite field

The finite field with p^n elements is denoted $GF(p^n)$ and is also called the Galois Field, mathematically

$GF(p^n) = \mathbb{Z}_p[x]/irreducible\ polynomial\ of\ degree\ n$

2.1.9 Residue classes

The set of elements (such as integers) that leave the same remainder when divided by a given modulus. i-e:

$[r] = \{x : x \in \mathbb{Z}\ and\ x \equiv r\ (mod\ m)\}$

2.1.10 Congruences

Let m be a fixed positive integer. Then an integer a is said to be congruent to an integer b modulo m, if m divides $a - b$. If a is congruent to b modulo m, we write $a \equiv b\ mod\ m$.

2.1.11 Residue class ring modulo m

The triplet $(\mathbb{Z}, +, .)$ is a commutative ring with unit element 1. This implies that $(\mathbb{Z}/m\mathbb{Z}, +, .)$ is a commutative ring with unit element 1 + $m\mathbb{Z}$. The latter ring is called the residue class ring modulo m.

Theorem

The residue class ring $\mathbb{Z}/m\mathbb{Z}$ is a field if and only if m is a prime number.

2.1.12 Order of a group

Let $a \in G$ and $a^n = e$ then order of a is n, where e is the identity of G and n is least power.

2.2 Cryptology

It is a science of coding and decoding secret messages. It can be subdivided into two disciplines.

(1) Cryptography (2) Cryptanalysis

2.2.1 Cryptography

A method in which we convert the data by encoding in such a form which cannot be readable.

Encryption and Decryption

"Encryption is that process in which we convert a messages (or plaintext) into an unreadable form (called ciphertext) and decryption is the process in which convert a ciphertext is again into plaintext.

An encryption scheme or cryptosystem is a tuple (P, C, K, E, D), where the following conditions are satisfied:

(1) P is a set of plaintext.

(2) C is a set of ciphertext.

(3) K is a set of keys.

TABLE 2.1 Correspondence between letters and numbers.

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

(4) $E = \{E_k, : k \in K\}$ is a family of functions $E_k : P \longrightarrow C$. Its elements are called encryption functions.

(5) $D = \{D_k : k \in K\}$ is a family of functions $D_k : C \longrightarrow P$. Its elements are called decryption functions.

(6) For each $e \in K$, there is $d \in K$ such that $D_d(E_e(p)) = p$ for all $p \in P$".

2.2.2 Goals of Cryptosystems

Confidentiality: Confidentiality is the characteristics of resource ensuring access is restricted to only permitted users, applications, or computer systems. Confidentiality deals with keeping information, networks, and system secure from unauthorized access.

Authentication: It is a mechanism to prove the identity of the sender that whether the message is incoming from the person is authorized or not, or fake identity.

Integrity: Integrity is defined as the consistency, accuracy, and validity of data or information. Integrity seeks to ensure that data is protected against any unauthorized or accidental changes.

Availaibility Availaibility describes a resource being accessible to a user, application, or a computer system when required. In other words the availaibilty means that when a user needs to get to information, he or she has the ability to do so.

Non-Repudation: It is a mechanism to prove that wheather the message was

really sent by the sender or not. So if the sender refuse that he could not send any message; this method not allows doing such type of action to sender.

2.2.3 Types of Cryptography

There are two main types of cryptography.

(1) Symmetric or Private key cryptography

(2) Asymmetric or public key cryptography

Private Key Cryptography

In which both the sender and reciever uses the sam key to encrypt and decrypt the messages.

It is divided into block cipher and stream cipher.

Block Cipher Block cipher is a method in which the plaintet is encrypted block wise.

Modes of Block Cipher

• There are four modes of operations:

(1) ECB mode

(2) CBC mode

(3) CFB mode

(4) OFB mode

ECB mode

$(e; Encryption)$

$(E_e; Encryption\ function)$

$(d; Decryption)$

$(D_d; Decryption\ function)$

$(m_j; plain - text\ j^{th}\ block,\ \ 1 \leq j \leq n)$

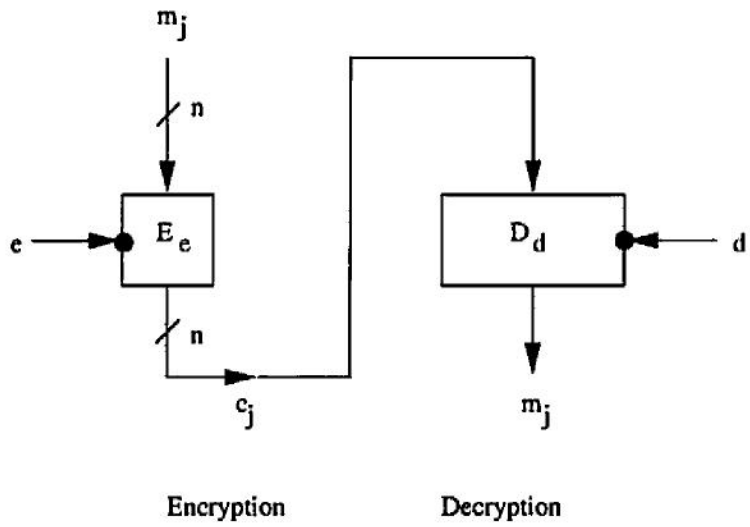

Figure 2.1:

$(n; total\ length\ of\ plain-text)$

$(c_j; cipher-text j^{th}\ block)$

CBC mode

$(e; Encryption)$

$(E_e; Encryption\ function)$

$(d; Decryption)$

$(D_d; Decryption\ function)$

$(m_j; plain-text\ j^{th}\ block,\ \ 1 \leq j \leq n)$

$(n; total\ length\ of\ plain-text)$

$(c_j; cipher-text\ j^{th}\ block)$

$(IV; Initialization\ vector)$

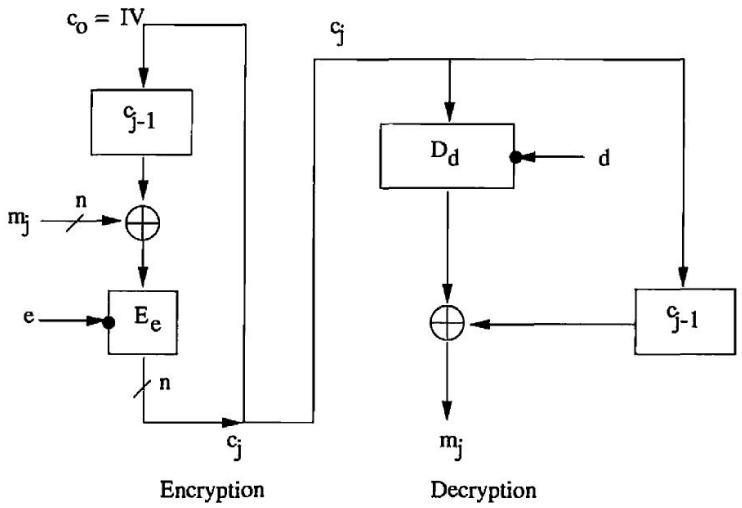

Figure 2.2:

CFB mode

$(K; key\ space,\ \ where k \in K)$

$(E_k; Encryption\ function)$

$(D_d; Decryption\ function)$

$(m_j; plain-text\ j^{th}\ block,\ \ 1 \leq j \leq n)$

$(n; total\ length\ of\ plain-text)$

$(c_j; cipher-text\ j^{th}\ block)$

$(IV; Initialization\ vector)$

$r; positive\ integer$

and for $1 \leq j \leq u$:

1. $O_j = E_k(I_j)$

2. Set t_j to the string, which consists of the first r bits of O_j.

3. $c_j = m_j \oplus t_j$.

4. $I_{j+1} = 2^r I_j + C_j\ mod\ 2^n$, so I_{j+1}

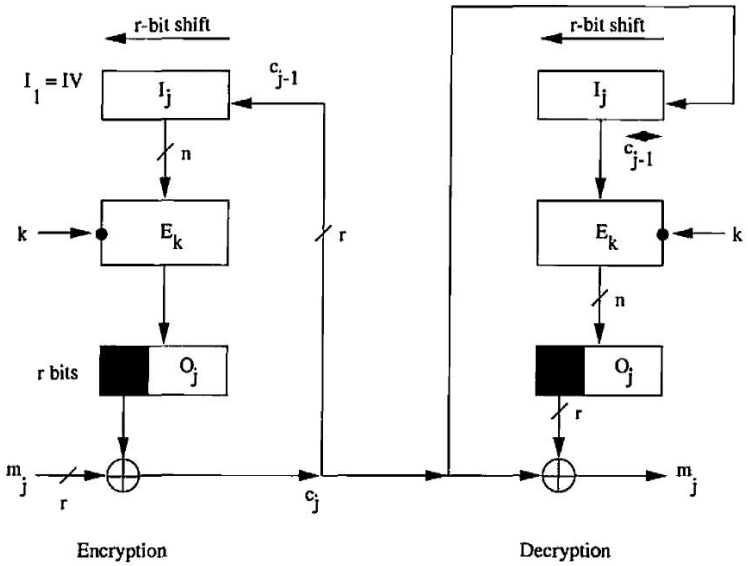

Figure 2.3:

OFB mode

$(K; key\ space,\ \ where k \in K)$

$(E_k; Encryption\ function)$

$(D_d; Decryption\ function)$

$(m_j; plain - text\ j^{th}\ block,\ \ 1 \leq j \leq n)$

$(n; total\ length\ of\ plain - text)$

$(c_j; cipher - text\ j^{th}\ block)$

$(IV; Initialization\ vector)$

$r; positive\ integer$

1. $O_j = E_k(I_j)$.

2. $I_{j+1} = O_j$.

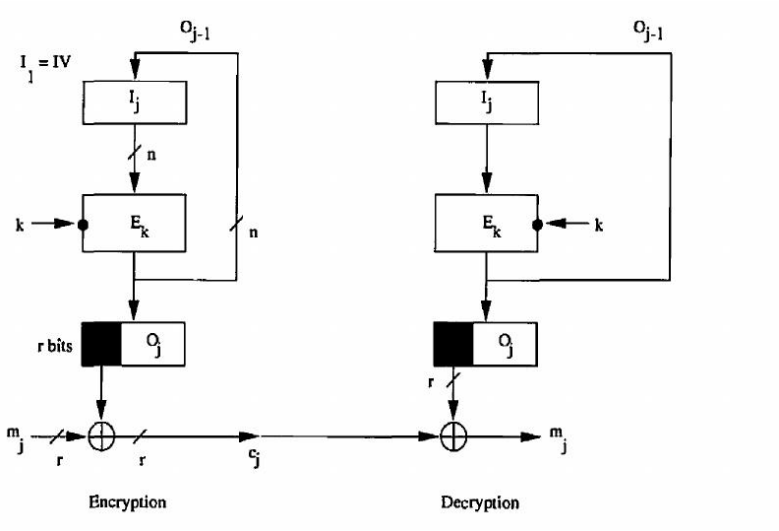

Figure 2.4:

Stream Cipher

Stream cipher is a particular form of block cipher in which length of a block is equal to 1. i-e the plain-text is encrypted bit wise.

$z_i = k_i, 1 \leq i \leq n, whereLetk = (k_1, ..., k_n$ and for $m > n$

$$z_i = \sum_{j=1}^{n} c_j z_{i-j} mod 2, \quad n < i \leq m,$$

$$z_{i+4} = z_i \oplus z_{i+1}$$

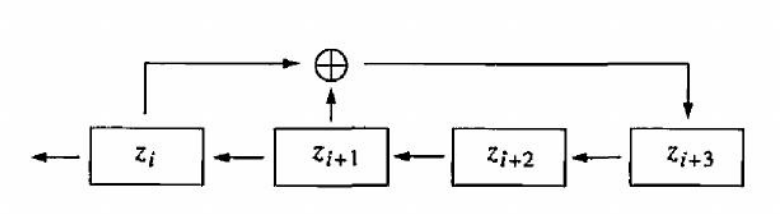

Figure 2.5:

Affine Cipher

$\Sigma = plain - text\ space = cipher - text\ space$

Let $\mathbb{Z}_m$ is plain-text and $\mathbb{Z}_m^2$ key space in which $(m, a) = 1$ for the pair $(a, b) = k$

Encryption

$$E_k : \Sigma \longrightarrow \Sigma, \quad x \mapsto ax + b\ mod\ m$$

Decryption

$$D_k : \Sigma \longrightarrow \Sigma, \quad x \mapsto \acute{a}(x - b)\ mod\ m$$

Public Key Cryptography

In which pair of keys (e, d) is used where e is an encryption key which is publically known and d is decryption key which is kept private.

RSA Cryptosystem

The first PKC was proposed by the mathematician Rivest, Shamir and Adleman, called the RSA cryptosystem. RSA is the number theoretic based encryption in which we take $n = pq$ where p,q are kept secret and n is the public key which is known publically and is being chosen the product large prime numbers. The ciphered text $c = x^e\ mod\ n$ and plaintext or original message x is obtained $x = c^d\ mod\ n$. The keys e and d are generated by using the little Fermat's Theorem as follows

• Compute $n = pq$ by taking two random large primes equal in magnitude and different in binary representation to make harder to guess.

• Compute $m = lcm(p-1, q-1)$ kept it secret.

• Choose e such that $gcd(e, m) = 1$, $1 < e < m$

Determine integer d such that $ed \equiv 1 \ mod \ m$ or $d = e^{(-1)} \ mod \ m$ the modular multiplicative inverse.

Diffie-Hellman In this protocol the secret key are exchanged between the parties over the insecure channel it based on the Elgamal in which the key is exchanged by using the private key system over insecure channel.

Example: Take $\mathbb{Z}/n\mathbb{Z}$ additive group for a positive integer n. The generator of this group is $1 + n\mathbb{Z}$. Let a of $b + n\mathbb{Z}$ to the base $1 + n\mathbb{Z}$ satisïňĄes the congruence $b \equiv a \ mod \ n$ Hence, $a = b$. Let $g + n\mathbb{Z}$ be the another generator of $\mathbb{Z}/n\mathbb{Z}$ with gcd$(g, n) = 1$. The discrete logarithm a of $b + n\mathbb{Z}$ to the base $g + n\mathbb{Z}$ satisïňĄes the congruence $b \equiv ga \ mod \ n$.

we use extended euclidean algorithm to solve the above congruences.

Structure of ElGamal

ElGamal has three phases:

(1) Structure

(2) Encryption

(3) Decryption

Structure

This is done by the party interested in receiving messages and this phase is done only once.

Let us suppose the receiver is (Bob), so Bob first chooses a large prime p and also chooses in $\alpha \in \mathbb{Z}_p^*$, where α is the generator and $\mathbb{Z}_p^*$ is the multiplicative cyclic group, where $p = 2q + 1$ and q is prime.

Now Bob chooses b, $0 < b < p$ with $(b, p-1)$ as his private key. Now Bob computes $\beta \equiv \alpha^b \ mod \ p$. So at last the Public key is $Kpub = (p, \alpha, \beta)$, so this key

publishes by the Bob once. Therefore anyone can send him secret messages.

Encryption

This is done by the party who wants to send the messages. This phase is executed every time a message is sent. Let us consider the sender is (Alice), if Alice wants to send a plaintext x to Bob, first Alice convert that plaintext into cipher text, so for this Alice chooses a, $0 < a < p$ with $(a, p-1)$. Now Alice computes the ephemeral key that is $K_E \equiv \alpha^a \ mod \ p$. Any time when Alice wants to send a message to Bob " K_E" would be change every time, Where "K_E" is also called Temporary key.

Now Alice computes the "shared key" that is $k \equiv \beta^a \ mod \ p$ and at last the plain text changes into cipher text by computing $y \equiv xk \ mod \ p$. Now if Alice, wants to send several messages, then she should chooses a at random for every message while conditions for choosing a remain the same.

Decryption

In this phase Bob gets (K_E, y) from Alice. Since Bob has his secret key b, and shared key that is $k \equiv K_E^b modp$, at last Bob gets the plain text by computing $x \equiv yk^{-1} modp$.

Outcome

(1) The protection of voter's intention by strong encryption along the path from voting client to the tally and the post-election process.

(2) The solid evidence that the vote is casted as it was intended.

(3) Election process can be verified by independent verifier. The verification includes the checks by election authority that the vote only casted by eligible voters.

(4) The individual anonymization or decryption key is shared between trust-

worthy group (the voter, election authority, independent election observer) to prevent the single point of failure.

Toy Example

We will illustrate ElGamal Encryption through an example below:

Bob chooses a prime p that is $p = 79$.

Now Bob chooses a generator of $\mathbb{Z}_{79}^*$ that is $\alpha = 30$. Now Bob chooses his private key that is $b = 61$. Now Bob computes $\beta \equiv \alpha^b \; mod \; p$ that is $\beta \equiv 30^{61} \; mod \; 79$. Hence we get $\beta = 59$. So the Bob Public Key is $Kpub = (79, 30, 59)$.

Now suppose Alice wants to send the plaintext $x = 44$ and Alice chooses a random number $a = 4$, so to convert the plain text into cipher text Alice computes $K_E \equiv \alpha^a \; mod \; p$ that is

$K_E \equiv 30^4 \; mod \; 79$ and hence computes $K_E = 13$. Now Alice compute the shared key; $k \equiv \beta^a \; mod \; p$ that is $k \equiv 59^4 \; mod \; 79$ and hence $k = 25$. Now for cipher text $y \equiv x \times k \; mod \; p$ that is

$y = 44 \times 25 \; mod \; 79$

$y = 25$

Bob received $(K_E, y) = (13, 73)$ Now Bob compute the shared key that is

$k \equiv 13^{61} \; mod \; 79$ and hence $k = 25$.

Finally to decrypt the original text Bob compute $x \equiv y \times k^{-1} \; mod \; p$ that is To find inverse of k Bob computes $k \times z \equiv 1 \; mod \; 79$ and at last

$x \equiv 73 \times 19 \; mod \; 79$ and thus $x = 44$

2.3 Cryptanalysis

It is basically the attacks on cryptosystems, we discuss some of the attacks below.

Types of Cryptanalysis

Plaintext-based Attacks	Ciphertext-based Attacks
Known Plaintext	Ciphertext only
Chosen Plaintext	Chosen Ciphertext
Adaptive Chosen Plaintext	Adaptive Chosen ciphertext

Known Plaintext and Ciphertext-only Attacks:

The plain-text is known by the attacker and the resulting cipher-text or some such type of pairs, this allows the attacker to analyze the relationship between the plaintext and ciphertext or even better the key.

The attacker knows ciphertexts of many messages, encrypted with same key, he tries to find the plaintext of these messages or even better the key.

Chosen Plaintext and Chosen Ciphertext Attacks:

The attacker can choose random plaintexts to be encrypted and obtain the corresponding ciphertext. The goal of the attack is to gain some further information which reduces the security of encryption.

The attacker gathers information, at least in part, by choosing a ciphertext and obtaining its decryption under an unknown key.

Adaptive Chosen Plaintext and Adaptive Chosen Ciphertext Attacks:

The attacker is able to encrypt plaintexts. He is able to choose new plaintexts as a function of the ciphertexts obtained under an unknown key. He tries to find the key used or to decrypt other ciphertexts.

Chapter 3

Electronic Voting based on Homomorphic encryption

3.1 Introduction

In this chapter, we will be studying, electronic voting (e-voting), receipt-free e-voting in particular. In electronic voting, the Public Key Cryptography (PKC), along with Certificate Authority (CA) for bind, public keys or Identity Based Entity (IBE) are used.

Our study is focused on IBE which is not required CA. Elgamal scheme which is based on cyclic group, initially, we will be studying the definitions directly used in e-voting then implementation of Elgamal in e-voting.

3.2 Homomorphic encryption

The computation on the ciphertexts are allowed through homomorphic encryption which then generate an encrypted data which, when decrypted as per requirement.

"It must matches with the result of the operations if they had been performed on the plaintext. The purpose of homomorphic encryption is to allow computation on encrypted data.

Without having access to the unencrypted data" the computations can be performed through cloud computing on homomorphically encrypted data. For different services without exposing the important data homomorphic encryption can be used to securely chain them together.

Malleability

In this techniques if it is possible for an attacker to transform a ciphertext into another ciphertext which decrypts to a related plaintext. That is, given an encryption of a plaintext x, it is possible to generate another ciphertext which decrypts to $f(x)$, for a known function f, without necessarily knowing or learning

x. Malleability is often an undesirable property in a general-purpose cryptosystem, since it allows an attacker to modify the contents of a message.

Cloud computing

In this computing paradigm a large number of systems in a pool are connected for public and private networks.

3.2.1 Partially homomorphic cryptosystems

In the following examples, the notation $E(x)$ is used to denote the encryption of the message x.

Implementation of Homomorphic Encryption

Unpadded RSA

Let the modulus m be the public key of RSA cryptosystem public and e be the exponent, then a message x is encryoted in the the following way $E(x) = x^e \ mod \ m$. The homomorphic property is then $E(x_1) \cdot E(x_2) = x_1^e x_2^e \ mod \ m = (x_1 x_2)^e \ mod \ m = E(x_1 \cdot x_2)$

ElGamal

In the ElGamal cryptosystem, suppose G is a cyclic group and let the order is q with generator g, also let (G, q, g, h) be the public key, where $h = g^x$, and x be the secret key, then a message m is encrypted as $E(m) = (g^r, m \cdot h^r)$ for some random $r \in \{0, \ldots, q-1\}$. The homomorphic property is then

$E(m_1) \cdot E(m_2) = (g^{r_1}, m_1 \cdot h^{r_1})(g^{r_2}, m_2 \cdot h^{r_2})$
$= (g^{r_1+r_2}, (m_1 \cdot m_2) h^{r_1+r_2}) = E(m_1 \cdot m_2).$

Goldwasser-Micali

In the Goldwasser-Micali cryptosystem, let modulus m be the public key and let x be the quadratic non-residue , then the encryption of a bit b is $E(b) = x^b r^2 \ mod \ m$, for some random $r \in \{0, \ldots, m-1\}$. The homomorphic property is then

$E(b_1) \cdot E(b_2) = x^{b_1} r_1^2 x^{b_2} r_2^2 \; mod \; m = x^{b_1+b_2}(r_1 r_2)^2 \; mod \; m = E(b_1 \oplus b_2)$

$\oplus$ denotes addition modulo 2, (i.e. exclusive-or).

Benaloh

In the Benaloh cryptosystem, let modulus m be the public key and let g be the base with a blocksize of c, then then a message x is encrypted as $E(x) = g^x r^c \; mod \; m$ for some random $r \in \{0, \ldots, m-1\}$. The homomorphic property is then

$E(x_1) \cdot E(x_2) \, mod \, m = (g^{x_1} r_1^c)(g^{x_2} r_2^c) \, mod \, m = g^{x_1+x_2}(r_1 r_2)^c = E(x_1 + x_2 \, mod \, m)$

Paillier

In the Paillier cryptosystem, let modulus m be the public key and g be the base, then the encryption of a message x is $E(x) = g^x r^m \; mod \; m^2$ for some random $r \in \{0, \ldots, m-1\}$. The homomorphic property is then

$E(x_1) \cdot E(x_2) = (g^{x_1} r_1^m)(g^{x_2} r_2^m) \; mod \; m^2 = g^{x_1+x_2}(r_1 r_2)^m \; mod \; m^2 = E(x_1 + x_2)$

3.2.2 Fully homomorphic encryption

A cryptosystem in which the arbitrary computation on cipher-text are performed which enables the construction of programs for any desirable functionality, which can be run on encrypted inputs to produce an encryption of the result. In this encryption the program never required its original inputs it can be used an untrusted party without revealing its inputs and internal state. Fully homomorphic cryptosystems have great practical implications in the outsourcing of private computations, for instance, in the context of cloud computing.

3.3 Electronic Voting (e-voting)

The properties of suggested eletronic voting protcols, used currently, based on Public Key Cryptography (PKC), which is the most suitable for key agreement protocols and authentication mechanisms. In PKC, the pair of encryption key

e which is known to public and the decryption d which is the secret or private key, are used. The voters use the key e to cast the vote and use key d to verify his/her vote being casted to intended candidate. Certification Authority (CA) is essential to implement. It gives certificates that bind public keys to entities, and enable their verification while using PKC. An alternatively, Identity-Based Encryption can be used such a way that we can have all the benefits offered by PKC, for which we no need to have the certificates and main components of a public key Infrastructure (PKI). The first PKC was proposed (in 1976 and published first research article on RSA in 1978) by Rivest, Shamir and Adleman, called the RSA public key Cryptography based on $n = pq$, p and q are very large prime number. p,q were kept secret while n to be publically known, the cipher text $c \equiv x^{pk} \; mod \; n$ and the plain text $x \equiv c^{sk} \; mod \; n$, the keys pk and sk are calculated by the little Fermat's Theorem. However, currently the ElGamal Encryption scheme is used being more robust. It operates on a cyclic group for which the decisional Diffi-Hellman assumption (DDH) holds. This group is the subgroup of Quadratic residues $G_q < \mathbb{Z}_p^*$ of prime order q, where $p = 2q + 1$ is a safe prime. The public parameters of an Elgamal Encryption scheme are thus p,q and a generator α excluding 1. The key pair β, b contain a random decryption key b and $\beta \equiv \alpha^b \; mod \; p$ is public encryption key. If x is plaintext and a a random value then $y \equiv xk \; mod \; p$, where $k \equiv \beta^a \; mod \; p$ is Elgamal Encryption. The decryption of a given encryption $e = (x, a)$ is $x \equiv yk^{-1} \; mod \; p$.

Keys generated by multiple parties can be combined to form a single public key. The corresponding private keys can be regarded as key shares and used in a collaborative manner. Being having mathematical background we will study the theoretical based cryptography. It is therefore, the following areas will be studied to be precised Elgamal encryption survey, consequently the implementation of modulo groups in cryptography, e-voting in particular.

3.4 Receipt free e-voting

“The receipt freeness prevent voters from proving their cast vote, and hence thwart vote-buying and coercion.

In other words, the voter should not be able to prove to a third party that he has cast a particular vote. He must neither obtain nor be able to construct a receipt proving the content of his vote”.

3.4.1 Mixnet Voting scheme

It is a receipt-free voting scheme by Lee et al. [3] addressing on removing user chosen randomness from ballot paper guraunteeing receipt-freeness.

“The voting stage consists of four sub-stages.

(1) Each voter prepares a first ballot by encrypting his vote. The ballot is then sent to tamper-resistant randomizer for randomization.

(2). The tamper-resistant randomizer randomizes the ïňĄrst ballot with re-encryption to produce a final ballot.

(3) The tamper-resistant randomizer also produces a Designated Verifier Re-encryption Proof to prove the correctness of re-encryption to the voter.

(4) The final ballot and the Designated Verifier Re-encryption Proof are then sent to the voter. Finally, the voter checks the Designated Verifier Re-encryption Proof, then signs and submits the final ballot if the check is accepted”.

3.4.2 Blind Signature voting scheme

It is an anonymous signature scheme. Use of the blind signatures protocol, designed and presented for the first time by Chaum [4], may be a solution of the

problem. The protocol's idea is as follows:

1. Voter create a message t and then encrypt it by using blind number r which gives $\acute{t}$ after encrypption.

2. $\acute{t}$ delivered to authorities for electronic signature which gives $\acute{p}$.

3. $\acute{p}$ is converted to $\acute{p}$ by voter.

4. $\acute{p}$ is the signature by voter on message t.

3.5 Implementation of ElGamal in e-voting

3.6 Structure of Protocol

Entities: let number of authorities $= N$ i-e $A_1, ..., A_N$ and number of voters $= M$. Set the threshold ,say, t which represents the authorities whhich is minimum in number that ensures to be honest during the whole protocol .

"**Communication:**The general public is communicated through the bulletin board displayed publically and that is every participant can write to (into his own section), but nobody can delete from. The bulletin board can be considered as public channels with memory. Furthermore, we suppose the existence of untappable one-way channels from the authorities to the voters. The security of these channels must be physical, in such a way that even the voter cannot demonstrate what was sent over the channel (of course, the voter can record all received data, but he must not be able to prove to a third party that he received a particular string). Even a coercer who is physically present at the voter's place must not be able to sniffering the untappable channels. Note that some physical assumption seems to be inevitable for achieving receipt-freeness. 2 Indeed, untappable one-way channels from the authorities to the voters (as assumed in [6]) are the weakest physical assumption for which receipt-free voting protocols are known to exist.

Key Management (Infrastructure): For each voter there is public key for each secret key, whereever it should be ensured that each voter aware oh his own secret key consistent with his public key. This assumption is usually used for voter identification in any voting protocol. For the aim of receipt-freeness, the knowledge of his own key is necessary. If a voter will be able to prove that he does'nt recognize his own key, then he will get a receipt (this holds for this protocol moreover as for the protocol in [6]). We suppose that the underlying public-key infrastructure ensures that each voter knows his own key, but nevertheless we present a verification protocol. If a voter exposes his own secret key to vote-buyer but still the property of receipt-free is achieved.

Generality: The protocol could be a 1-out-of-L voting scheme, where ever each entitled voter could submit one vote from the set V of valid votes, $|V| = L$ $(e.g.\ L = 2\ and\ V = \{1, 1\})$. The objective of this protocol is to calculate the tally as the sum of the cast votes securely. Note that the restriction on a distinct set of valid votes is important in any receipt-free voting scheme. Voting schemes that allow the voter to cast an arbitrary string as his vote cannot ensure receipt-freeness, because they allow the voter to tag his vote".

Security/Secrecy:

The security of this protocol depend upon the correct execution of the computed tally,i-e. During whole protocl execution at least t authorities must be remain honest. (correctness); No group of less that t authorities can decrypt the casted vote (privacy); and

"that the voter will not be able to prove to a vote-buyer or coercer a casted vote".

"(receipt-freeness). In general, we suppose that authorities do not collude with the vote-buyer, respectively the coercer. However, under certain conditions some colluding can be tolerated."

3.7 Protocol Overview

"First of all each valid vote is encrypted by some deterministic way (e.g., by using the encryption function with 'randomness' 0). The list of such votes is to be known publically (on the very left in the figure). Then, the first authority picks this list, shuffles it, and hands it to the next authority. To shuffle the list means to re-randomize each entry" and permutation are made to reorder the entries. In the figure, drawing a circle around the secret denotes the encryption, and rotating the secret dentes the re-randomization. After that the next authority chooses the list, shuffles it, and so on. In addition to this shuffling, each authority should secretly exposed a secure untappable channel(that cannot be recorded). "This permits the voter to keep the track record of the ordering of the encrypted entries, and once each authority has shuffled the list, he can point to the encrypted vote of his choice". If a voter is colluding with an authority then for the prevention of casting an invalid vote, the list of the vote must be correctly shuffled by each authority and prove it publicly (without exposing the re-ordering).

"Vote casted in this way is called receipt-free: beacause of its private verifiability of all the shuffling was done, the voter has no way to convince the third party of the content of his vote". To achieve This pwerful property we will use designated-verifier proof technique [7].

3.7.1 Requirements for the Basic protocol

A voting protocol which is based on homomorphic encryption is supposed to be a non-receipt free, to make it a receipt- free voting scheme, we have to required some more properties of its encryption function. An probablistic encryption function is denoted by E, where set of encryption function for a vote v is denoted by $E(v)$. If e is the encryption of vote v, then it is a particular encryp-

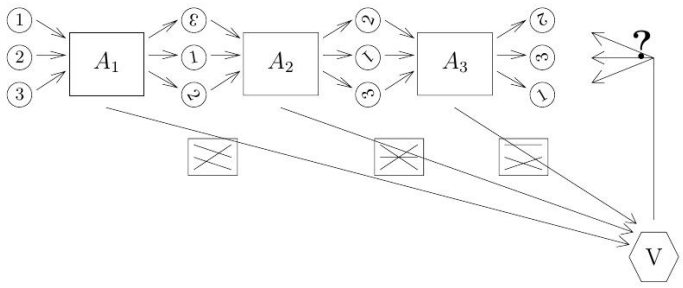

Figure 3.1:

tion of V i-e $e \in E(v)$. For non-receipt free voting protocol we just required 1-3 properties while for a receipt free voting protocol we required 4-6 properties.

1. **Secrecy of Encryption**

No one can decrypt any encryption if the group of authorities is less than t.

2. **Homomorphic Property**

"We suppose that the encryption function is homomorphic, that is, given the encryptions $e_1 \in E(v_1)$ and $e_2 \in E(v_2)$, the addition of these encryptions gives an encryption $e = e_1 \oplus e_2$ that encrypts the sum vote, i.e. $e \in E(v_1 + v_2)$. We need this addition can be done efficiently without any secrets".

3. **Verifiable Decryption**

The sum of votes which is encrypted requires an efficient protocol for verifiablity decrypting. Let T be the sum of votes, then the authorites provides the T (i-e sum of vootes) for any given encryption $e \in E(T)$ and also provides e that must decrypts to T. Both the proof and decryption must also work if $N - t$ authorities refuses cooperation but the still the decryption and proof works. No any information can be revealed that weaken the property 1 (secrecy) of other encryption.

"4. **Random Re-encryptability**

We need an algorithm for the random and arbitrary re-encryption of any encryption e.

Given $e \in E(v)$ (where typically v is unknown), there is a probabilistic re-encryption algorithm R that outputs $\acute{e} \in E(v)$, where $\acute{e}$ is uniformly distributed over $E(v)$. We call the randomness used for generating $\acute{e}$ the witness".

"5. **1-out-of-L Re-encryption Proof**

This property is based on random re-encryptability. Let $e_1, ..., e_L$ be the list of encryptions. We need an efficient protocol that for any given encryption e, we have witness that e_i is a re-encryption of e (for a given i), proves that indeed e_i is a re-encryption of e, without exposing i. This proof is known as 1-out-of-L re-encryption proof.

6. **Designated-Verifier Re-encryption Proof**

We suppose the existence of an efficient protocol that given encryptions e and $\acute{e}$ and a witness for $\acute{e}$ being a re-encryption of e, the existence of this witness will be proved in such a way that the correctness will be only verified by the designated verifier [7]. This proof is known as designated-verifier re-encryption proof."

Introducing Receipt-freeness

We first show how authorities generated the votes and how the vote is casted by the voter, then how the tallying is done.

Vote Generation: Without loss of generality, suppose that there is a standard encryption $e_i^{(0)}$ for each valid vote $v_i \in V$, it is clear that which v_i a given encryption $e_i^{(0)}$ belongs to a technique where encrypted votes are generated by using the probabilistic encryption technique E which gives the randomness of all-0 string. This encrypted vote can be decrypted simply by trying all valid votes $v \in V$ Hence, the list of the standard encrypted vlaid votes is $e_1^{(0)}, ..., e_L^{(0)}$ which is publically known. In turn, for each authority A_k $(where\ k = 1, ..., N)$:

"1. A_k picks the list $e_1^{(k1)}, ..., e_L^{(k1)}$ of encrypted valid votes (for the ïňĄrst authority A_1, this is often the general public list of standard-encrypted valid votes, and for all succeeding authorities, this is often the list of the previous authority).

Then the authority shuffles this list randomly, and hands it to the succeeding authority. To shuffle the list means to re-encrypt each encrypted vote (Property 4)" and permutations are made to reorder the list. More precisely, the authority randomly selects a permutation $\pi_k : \{1, ..., L\} \rightarrow 1, ..., L$, computes a random re-encryption of $e_i^{(k1)}$ and assigns it to $e_{\pi_k(i)}^{(k)}$ $(forall i = 1, ..., L)$.

2. A_k shuffled the lisy honestly and this would be proved publically, namely by proving for each i, there exists a re-encryption of $e_i^{(k1)}$ in the list $e_1^{(k)}, ..., e_L^{(k)}$ without exposing which (1-out-of-L re-encryption proof, Property 5).

3. The permutation π_k which is used for the reordering of encrypted votes would be transmit secretly to the voter by an authority A_k and proves its correctness privately. More precisely, the permutation π_k and a designated-verifier proof for each $i = 1, ..., L$, that $e_{\pi_k(i)}^{(k)}$ is a re-encryption of $e_i^{(k1)}$ (Property 6), is sent to the voter by untappable channel.

4. If the proof is not acceptable by the voter, and publicly complains against the authority. In this situation we set $e_1^{(k)} = e_1^{(k-1)}, ..., e_L^{(k)} = e_L^{(k-1)}$, i.e. the reordering of that authority is ignored. The complain against the authorities by the voter would be at most $N - t$.

"**Casting a Vote:** The voter derives the position i of the encrypted vote $e_i^{(N)}$ of his choice, and publicly announces it.

TALLYING: The chosen encrypted votes of all voters are then summed for tallying. More precisely, they are added (using homomorphic addition $\oplus$, Property 2) to achieve an encryption $E(T)$ of the sum T of the votes. The authorities decrypt and output T and prove its correctness (Property 3)".

3.8 Security

"**Correctness:**The tally correctness is guaranteed if all voters can cast the vote they wish(i.e.can trace the permutations of the authorities,Property6), if the votes cannot be casted as an invalid vote (Property 5), if the encrypted sum can b computed publically correct (Property 2), and if the decryption of the sum is verifiable (Property 3)".

"**Privacy:** The privacy of authority is assured if an outstanding person or group of less that t authorities cannot decrypt the encrypted vote (Property 1). Also, given a list of encrypted votes and a shuffled list, it must be infeasible to find out which vote in the original list was permuted to which vote in the shuffled list (Property 4). As at least t shufflings are done correctly (at most $N - t$ shufflings will be ignored by a complaining voter), group of less than t authorities would not b able to ïňĄnd out the reordering of the list".

"**Receipt-Freeness:** The voter come across the two stages during vote casting the first one is a vote generation where voter can disable the shuffling of upto N-t authorities. In second stage which is vote casting, the voter select the encrypted vote of his choice through the untappable channels, the voter receives the permutations π_k and the designated-verifier proofs for the correctness of each π_k. Because of non-transferability of designated-verifier proofs (Property 6) and untappability of the channel used the voter may refuse for the usage of any permutation π_k, which is enough for not being able to prove the cast vote. In this scheme a coercer can coerce a voter rather a vote or can coerce voter to vote randomly, though this scheme is receipt-free e voting scheme.

In another situation when authorities unite with another illegal entity with coercer (vote-buyer) then the receipt-freeness seems still ensured, apparently as long as each voter knows at least one authority which does not collude with the vote-buyer (then the voter can lie for the permutation π_k of this authority A_k).

In another case If a voter does not know about that authority, he can randomly select one authority and lie for this permutation. In the scenario of vote-buying the voter can forge a receipt for a vote which is not casted by him, and also such a forged receipt will be accepted by the vote-buyer with probability linear in the number of authorities not colluding with him, that looks to be unacceptable for the vote-buyer. However, in the the context of coercion, the probability of a lying voter to be caught is linear in the number of authorities colluding with the coercer, and this looks to be unacceptable for the voter.

3.9 Made Receipt-Free

We studied that the foundation of receipt-free 1-out-of-L voting scheme is the construction and protocol of Cramer, Gennaro, and Schoenmakers [9].

3.9.1 Homomorphic ElGamal Encryption

This encryption scheme is similar to the scheme used in [9]". The summary of this scheme is given brief firstly. The foundation of this scheme is ElGamal cryptosystem [12]. Let $|G| = q$, where q is a huge prime. G is a subgroup of $\mathbb{Z}_p*$, p is a huge prime, where G is a commutative subgroup of $\mathbb{Z}_p*$. Let the generator of G is g i.e. $G =< g >$. Let z be the secret key whic is chosen uniformly from $\mathbb{Z}_q$, where $h = g^z$ is the public key. The key pair (z, h) share with the authorities in such a way that each autority receive their own secret share according to (t, n) threshold secret sharing scheme. Each share is given by $h_i = g_i^z$. Now let γ be another generator of G, where as V is the set which contains of valid votes in $\mathbb{Z}_q$. The vote $v \in V$ is to be encrypted is given by

$$E(v) = (g^\alpha, \gamma^v h^\alpha)$$

where $\alpha \in R\mathbb{Z}_q$ is a random number and γ^v is the message in the context of ElGamal.

“

3.9.2 Encoding of Votes

There are multiple ways of encoding L votes in $\mathbb{Z}_q$, so that sum of several votes gives the sum of each type of vote.

Example :

Let $L = 2$, take $V = \{+1, 1\}$ implies how many $+1$ votes and how many 1 votes were cast from the sum of votes. For special cases when $L > 2$, one can use the said approach yet.

For example: If voters have the options to cast yes, no, or empty. Our interest is to count the number of yes and number of no's and check weather there are more yes or more no. Regardless of number of empty votes. This fact can be encoded as +1 for yes, 1 for no and 0 for empty. It should be possible to find the exact number of cast votes for each choice. Along the ideas of [8], one can set $V = 1, M, M^2, ..., M^{L-1}$, where M is the the number of voters. One can easily compute the number of cast votes for each choice, once the sum of the votes is computed”.

3.9.3 Main properties of protocol

1. **(Secrecy)** As z is the secret key so it must be shared among the authorities according to (t, n) threshold secret sharing. i-e any group of authorities which is less than $t - 1$ will not b able to compute z. If the secrecy of the scheme is violated then it means that either ElGamal [12] is broken or the secret-sharing scheme [11].

"2. **(Homomorphic Property).**If an encryptions $e_1 = (x_1, y_1)$ and $e_2 = (x_2, y_2)$ then their addition is defined as

$$e_1 \oplus e_2 = (x_1 x_2, y_1 y_2).$$

Since $e_1 \in E(v_1)$ and $e_2 \in E(v_2)$, so $(e_1 \oplus e_2) \in E(v_1 + v_2)$.

3. **(Verifiable Decryption.)**. If we want to decrypt T from $e = (x, y)$ first all the authorities compute togetherly" $\hat{x} = x^z$. To prove $\hat{x} = x^z$ for this all the authorities A_i must compute $\hat{x}_i = x_i^z$, where z_i is $A_i s$ share of secret share of z, and then $\hat{x}$ can be computed from $\hat{x}_i$. This can be only done if atleast t autohrities prove $\hat{x}_i$. When $\hat{x}$ is computed the we can easily compute

$$\frac{y}{\hat{x}} = \frac{\gamma^T \cdot h^\alpha}{(g^\alpha)^z} = \gamma^T$$

Where T must be find by the authorities.

4. **(Re-encryptability).** Let $\acute{e} = (\acute{x}, \acute{y})$ be the re-encryption of an encrypted vote $e = (x, y)$

$$(\acute{x}, \acute{y}) = (g^\xi x, h^\xi y)$$

for a random integer $\xi \in R\mathbb{Z}_q$.

5. **(1-out-of-L Re-encryption Proof)**. Let e be the encrypted vote and $e_1, ..., e_L$ be the list of encrypted votes which contained e. In this list all the authorities proves the re-encryption of e.

6. **(Designated-Verifier Re-encryption Proof)**. In this all the authorities proves in privately manner than an encrypted vote $\acute{e}$ is a re-encryption of e. For this we require an efficient indistinguishable protocol.

3.10 Concluding Remarks

Initially, we have studied, the basic construction of receipt-free protocol, we obtained more efficient receipt-free protocol after applying the generic protocol, called receipt-free 1-out-of-L voting scheme, to the multi-authority voting scheme, based on homomorphic ElGamal encryption.

Bibliography

[1] Adi Shamir, *How to share a secret. Communications of the ACM*, 22:612-613, 1979.

[2] Byoungcheon Lee, Colin Boyd, Ed Dawson, Kwangjo Kim, Jeongmo Yang, and Seungjae Yoo, *Providing receipt-freeness in mixnet-based voting protocols*. In Information Security and Cryptology-ICISC 03, pages 245-258, 2004.

[3] Craig Stuntz, *What is Homomorphic Encryption, and Why Should I Care?*, March 2010.

[4] Chaum, D, *Blind Signatures for Untraceable Payments*. In: Crypto'82, pp. 199-203. Plenum Press, New York (1983).

[5] Johannes A. Buchmann,*Introduction to cryptography*. ISBN 0-387-95034.

[6] Kazue Sako and Joe Kilian, *Receipt-free mix-type voting scheme - A practical solution to the implementation of a voting booth*. In Advances in Cryptology - EUROCRYPT '95, vol. 921 of LNCS, pp. 393-403. Springer-Verlag, 1995.

[7] Markus Jakobsson, Kazue Sako, and Russell Impagliazzo, *Designated verifier proofs and their applications*. In Advances in Cryptology - EUROCRYPT '96, vol. 1070 of LNCS, pp. 143-154. Springer-Verlag, 1996.

[8] Ronald Cramer, Matthew K. Franklin, Berry Schoenmakers, and Moti Yung, *Multi-authority secret-ballot elections with linear work*. In Advances in Cryptology - EUROCRYPT ?96, vol. 1070 of LNCS, pp. 72?83. Springer-Verlag, May 1996.

[9] Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers, *A secure and optimally efficient multi-authority election scheme*. European Transactions on Telecommunications, 8:481?489, 1997. Preliminary version in Advances in Cryptology - EUROCRYPT '9.

[10] Ron Rivest, *Voting, Homomorphic Encryption (PDF)*, Oct 2002.

[11] Torben P. Pedersen, *A threshold cryptosystem without a trusted party (extended abstract)*. In Advances in Cryptology - EUROCRYPT '91, vol. 547 of LNCS, pp. 522-526. Springer-Verlag, 199.

[12] Taher ElGamal, *A public key cryptosystem and a signature scheme based on discrete logarithms*. In Advances in Cryptology - CRYPTO '84, vol. 196 of LNCS, pp. 10-18. Springer-Verlag, 1984.