An Implementation of a Secure Internet Voting Protocol

Beeshmoy Kumar Mohanty (10606004)

Sushil Deoghare (10606031)

Ezhilarasan M. (10606024)

Department of Computer Science and Engineering

National Institute of Technology Rourkela

Rourkela-769 008, Orissa, India

An Implementation of a Secure Internet Voting Protocol

A Thesis report submitted in partial fulfillment

of the requirements for the degree of

Bachelor of Technology

In

Computer Science and Engineering

By

Beeshmoy Kumar Mohanty (10606004)

Sushil Deoghare (10606031)

Ezhilarasan M. (10606024)

Under the guidance of

Prof. Sujata Mohanty

Department of Computer Science and Engineering

National Institute of Technology Rourkela

Rourkela-769 008, Orissa, India

Department of Computer Science and Engineering

National Institute of Technology Rourkela

CERTIFICATE

This is to certify that the work in the thesis entitled "**Secure Internet Voting Protocol**" submitted by Beeshmoy Kumar Mohanty, Sushil Deoghare and Ezhilarasan M. is a record of an original research work carried out by them under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of *Bachelor of Technology* in Computer Science and Engineering during the session 2008–2009 in the Department of Computer Science and Engineering, National Institute of Technology Rourkela. To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University / Institute for the award of any Degree or Diploma.

Date:

Prof. Sujata Mohanty

Dept. of Computer Science and Engineering

National Institute of Technology, Rourkela

ACKNOWLEDGEMENT

We express our sincere gratitude to Prof. Sujata Mohanty, Department of Computer Science and Engineering, National Institute of Technology, Rourkela, for her valuable guidance and timely suggestions during the entire duration of my project work, without which this work would not have been possible. We would also like to convey our deep regards to all other faculty members and staff of Department of Computer Science and Engineering, NIT Rourkela, who have bestowed their great effort and guidance at appropriate times without which it would have been very difficult on my part to finish this project work. Finally I would also like to thank my friends for their advice and pointing out our mistakes.

Beeshmoy Kumar Mohanty	Sushil Deoghare	Ezhilarasan M.
Rollno.10606004	Rollno.10606031	Rollno.10606024

TABLE OF CONTENTS

Table of figures:

ABSTRACT

Voting is one of the most important activities in a democratic society. In a traditional voting environment voting process sometimes becomes quite inconvenient due to the reluctance of certain voters to visit a polling booth to cast votes besides involving huge social and human resources. The development of computer networks and elaboration of cryptographic techniques facilitate the implementation of electronic voting. In this work we propose a secure electronic voting protocol that is suitable for large scale voting over the Internet. The protocol allows a voter to cast his or her ballot anonymously, by exchanging untraceable yet authentic messages. The e-voting protocol is based on blind signatures and has the properties of anonymity, mobility, efficiency, robustness, authentication, uniqueness, and universal verifiability and coercion-resistant. The proposed protocol encompasses three distinct phases - that of registration phase, voting phase and counting phase involving five parties, the voter, certification centre, authentication server, voting server and a tallying server.

Chapter 1

Introduction

1.1 Introduction

Election and voting are all now well known terms in modern days of Democracy. Stones and pot shards dropped in Greek vases led to paper ballots being dropped in sealed boxes. Nowadays, new technologies are developed to automate the voting process. The automation should preserve the security of the traditional elections (especially the privacy of the votes). Mechanical voting booths and punch cards have already been designed to replace paper ballots for faster counting.

Electronic online voting over the Internet would be much more profitable. Many voters would appreciate the possibility of voting from anywhere. Convenience of the voting will result in increasing the number of participating voters. Fast, cheap and convenient voting process could have great impact on the contemporary democratic societies. Electronic voting, as the name implies, is the voting process held over electronic media, i.e. computers. For such a sensitive issue like election, security is one of the main concerns. But simplicity is also necessary to ensure the participation of common people. Besides security and simplicity, there may be some other issues that need to be considered. In that respect, we need to specify all such issues or properties that the election system must possess. A well-defined protocol is necessary to take care of all such requirements.

Computers and facilitation of internet has spread its wings far and wide providing easy access everywhere. Thus in this context, holding election over the Internet seems logical from many different points of view. Relief from long queues, minimal chance of voting error, verifiability (not possible in case of face-to-face service) stands in favour of an electronic election. Recent improvements in network security have made it possible to design election system with high class security. But it is also important that carefully designed protocols and continuous improvements of the implementations are necessary to keep them out of reach from the network threats. From that point of view, an implementation of secure Internet voting protocol appears to be another application of cryptography and network security.

Electronic voting has been intensively studied for over the last twenty years. Many e-voting protocols, therefore, have been proposed in the last several decades and both the security as

well as the effectiveness has been improved. Nevertheless, to the best of our knowledge, no practical and complete solution has been found for large scale elections over a network, say Internet. This paper suggests a practical application of the existing cryptographic schemes that ensures a fool-proof and verifiable protocol which can be implemented over the internet satisfying all e-voting security requirements.

Design of secure e-voting protocols over a network is indeed a very difficult task as all the requirements of the voting system have to be met. Failure to ensure even one of the specifications can lead to chinks and glitches that can be exploited by a middleman to forge or manipulate the intricate details.

The most efficient voting protocols could be categorized by their approaches into two major types: schemes using blind signatures and schemes using homomorphic encryption. The suitability of each of these types varies with the conditions under which it is to be applied.

In the schemes using blind signatures, the voter obtains a token – a blindly signed message unknown to anyone except himself. Next, the voter sends his token together with his vote anonymously over a secure channel. These schemes require voter's participation in more rounds.

In the schemes using homomorphic encryption the voter cooperates with the authorities to construct an encryption of his vote. Due to the homomorphic property, an encryption of the sum of the votes is obtained by multiplying the encrypted votes of all voters. Subsequently, the result of the election is computed from the sum of the votes which is jointly decrypted by the authorities.

A voting scheme must ensure that the voter can keep his vote private. In other words, the voter should not be able to prove to the third party that he has cast a particular vote. He must not be able to construct a proof of the content of his vote. This property is referred to as receipt-freeness. Only a few schemes guaranteeing receipt-freeness have been proposed. Among these schemes, receipt-free scheme using blind signatures assumes the existence of a special anonymous untappable channel. Achieving the communication that is both secure and anonymous, however, is extremely difficult.

1.2 Motivation and Objective

This project work is motivated by the fact that there needs to be an assurance of certain important issues that govern the basic principles of a secure voting protocol and the need to implement one based on existing cryptographic schemes for voting. Traditional techniques implemented for election and voting often suffered from drawbacks ranging from coercivity to anonymity. The basic idea of employing the blind signature schemes using a randomizing factor at each stage is to ensure that the manipulation of the vote is avoided, while blinding the vote and passing it over a secure channel and at the same time receiving the digital signatures from an authentication centre which vouches for the integrity of the votes and final unblinding the votes and tallying the results. Our implementation involves basically the aforesaid steps. An empirical study of the various existing protocols revealed the pros and cons of using each for assuring security in a voting system.

Chapter 2

Literature Study and Related work

2.1 Traditional Elections

The electronic voting system over the internet must emulate certain features of traditional voting schemes. We briefly sketch the most important ones of them.

Voting committee takes care of voters: It allows only eligible voters to vote, and ensures that every voter votes only once. After the elections, voting committee counts the votes and publishes the result. The ballots remain a secret, that is, a person's way of voting cannot be comprehended by a third party, the scheme just specifies that the person has voted and his name in the existing list of the voters is marked. Even if the voter tells his vote to another voter or a third party, he/she will not believe him – he can easily lie. On the other hand, the voter cannot be absolutely sure that his vote was really counted. He can just believe it was. Everybody has to believe that the voting committee is honest and it would not disrupt the elections.

2.2 The Basic Model

In the proposed model laid out for implementation, the parties involved are the voters, the authentication server, the certification server, the counting server and the voting server. The voter can be viewed as a client who connects to the server as a single thread; his casting of vote implicitly calculates the ballot value. The servers can be seen as authorities involved in a voting scheme with each server performing the stipulated work running parallel in the system.

2.2.1 The Voter

In general, voters are not willing to comply with complicated and time spending voting process. Therefore voter's actions and computations during the electronic voting should be kept at minimum, realizing vote-and-go concept. Voter can abstain from voting if he wishes to – he need not participate in the voting, or he can stop his voting any time before it is finished (of course, in this case his vote is not counted).

2.2.2 The Authorities

The authorities are the participating parties who manage the complete voting scheme. They have large computing power and they can store large amount of data in secret. It is assumed in the given scenario that the participating authorities perform the assigned work and neither of the authorities can be disruptive or revealing. Furthermore, they are expected to authorize the votes and provide a reliable scheme as a whole.

2.2.3 The Vote

The structure of the vote largely depends on the type of election system being followed, i.e., the mode of selection of candidates in the vote

The various voting schemes[14] are elucidated as follows:-

- One choice yes/no format – in this format the voter's choice is either a yes or no. Vote is generally one bit – 1 for yes and 0 for no.
- Single choice from n possibilities – This format has "n" choices of which the voter is expected to choose only one.
- K out of L voting – this formant has the voter selecting K choices out of L given choices. Vote here is a K-tuple ($v_1 \cdots v_K$).
- K out of L ordered voting – in this format the voter selects K choices and orders them unlike the previous case which only included random choicing. Vote here is an ordered K-tuple ($v_1 \cdots v_K$).
- Structured voting – In this format there are n levels of possibilities. Voter moves from the first level to the last one. At the i^{th} level he can select at most k_i possibilities from the subset Si of all possibilities in the i^{th} level.
- Write-in format – In this format the voter casts his own vote by writing his own answer wherein the vote is generally a string of specified length.

In addition to the voting formats discussed above there may be categorization on the basis of equality of votes. It can be equal-voting where the vote of the user is counted at most once else it can be calculated with a given weight w. This weight w is determined as a result of the priority and privilege of the voters participating. A structure containing the

vote is called a ballot. It can be easy, difficult or impossible to extract the vote from the ballot, depending on the scheme.

2.2.4 Communication Channel

Based on the type of communication being followed , the channels that define the method of communication between the voters and the authorities, that is the way the vote is channelized from the voter to the server, can be of three major categories- untappable channel , anonymous untraceable channel , untappable anonymous channel.

The channels must be fully secure and provide for efficient data communication minimizing network congestion and avoiding jitters if any.

2.3 Electronic Voting Scheme

Both the authorities and the voters have to follow electronic voting schemes which prescribe the voter's and the authorities' actions and computations during the period of voting. The voting is done with the voter as a client in the voting server, the vote of the user generating a message that is encrypted with a suitable key and sent across to the servers asking for signatures and certification. Finally the valid votes are updated into the database in the counting server and the results are determined. This section envisages the different phases in the voting system and the various trust requirements the protocol aims to fulfill.

2.3.1 The Phases of the e-voting scheme

The entire voting scheme is divided into a number of phases to simplify the actions and the procedures to be followed by clearly identifying the steps to be followed in each phase

- **Registration or Initialization Phase** - In this phase the voter registers himself with the system. Each registered user receives a unique user_id and a password assigned to him that uniquely identifies the voter, thereby preserving the identity of the voter. The

authorities setup the system, generate their public and secret keys and publish the public values.

- **Voting Phase** – In this phase the voters cast their votes. The voter communicates with authorities through the channels he can use, forming a ballot containing his vote. Finally he sends his ballot to its destination.
- **Counting Phase** - Authorities use their public and secret information to open the ballots and count the votes. They publish the result of elections.

2.3.2 The Security Requirements of an Efficient e-voting System

For the e-voting system to function properly that ensures error-free and robust electronic voting over the internet, it must satisfy the following criteria.[11]

- **Eligibility** - Only eligible voters can vote and no one votes twice
- **Anonymity** - Any traceability between the voter and his vote must be removed.
- **Verifiability** - A voter is able to verify that his or her vote is counted in the final tally. So also a passive observer can check that the election is fair: the published final tally is really the sum of the votes
- **Fairness** - No one should be able to compute a partial tally as the election progresses
- **Coercibility** - No one can use force or compel anybody to vote
- **Receipt-freeness** - A voter cannot prove that he or she voted in a certain way.
- **Privacy** - No coalition of participants (of reasonable composition) not containing voter himself can gain any information about the voter's vote. By reasonable composition we mean coalition of at most t authorities and any number of voters.
- **Robustness** - Faulty behavior of any reasonably sized coalition of participants can be tolerated. No coalition of voters can disrupt the election and any cheating voter will be detected.

2.4 Cryptographic Building Blocks

In this section we lay down some of the known Cryptographic concepts that are vital for the functioning of the electronic voting system.

2.4.1 A List of Notations used

$V_1, V_2, \ldots, V_M$	M voters
$v_1, v_2, \ldots, v_M$	votes of the voters
Z_p	field of positive integers modulo p, where p is prime number
Z_n	set of integers modulo n, i.e. $\{0, 1, \ldots, n-1\}$
Z_n^*	set of integers from Z_n relatively prime to n
Gcd (a, b)	greatest common divisor of the integers a, b
$a \oplus b$	bitwise exclusive or

2.4.2 Bit-Commitment Scheme

The bit-commitment scheme involves two parties – a sender (Alice) and a receiver (Bob). Suppose that Alice wants to send a message "m" in "b" bits to Bob and doesn't want to reveal "b" to Bob immediately. As per Bob, Alice should not be allowed to change her mind in the meantime and the bit she later reveals will be the same as she thinks of now. Alice encrypts the bit "b" and sends the encrypted bit to Bob. Bob, however, is not able to recover b until Alice sends him the key. Encryption of b is called a blob.

The bit commitment scheme is a function ξ: $\{0, 1\} \times X \rightarrow Y$,[14]

where X, Y are finite sets. An encryption of b is any value ξ (b, k), k $\in$ X.

PROPERTIES -

- Concealing – Bob cannot determine the value from the encrypted blob.
- Binding – Alice can then reveal the b,k used to construct the encrypted bit and open the blob

Hence, if Alice wants to commit to a string of bits, she commits each bit independently.

Now consider a variant of the bit commitment scheme, which we call trapdoor bit commitment[7]. Alice wants to commit two bits with Bob. Later, she wants to reveal only one of the committed bits to Bob. To implement this[], Alice sends [{b}k, $\{b_0\}k_0$] to Bob. If she wishes to reveal b, she sends ($\{b_0\}k_0$, inv(k)) to Bob. If she wishes to reveal b_0, she sends ($\{b\}_k$, inv(k_0)). It is easy to show that exactly one of the bits is revealed to Bob.

2.4.3 RSA Cryptosystem

The RSA Cryptosystem[2] consists of mainly the following phases:-

1. **Key Generation –** The sender Alice creates her public key and a corresponding private key and then follows the following steps :-
    - Generate two large random distinct primes p, q
    - Compute $n = pq$, $n' = (p - 1)(q - 1)$
    - Select an integer e, $1 < e < n'$ such that $\gcd(e, n') = 1$
    - Find an integer d, $1 < d < n'$ such that $ed \equiv 1 \bmod n'$
    - Thus, the public key is (n, e), the private key is d.
2. **Encryption** - To encrypt an integer m, $0 \leq m < n$, Bob, the receiver, should compute $c = m^e \bmod n$.
3. **Decryption** - Alice now computes the plaintext m from the cipher text c as

$$m = c^d \bmod n.$$

2.4.4 Digital Signatures

Digital Signatures, in the field of Cryptography and Network Security are generally, in layman terms, the computational analog of written signatures. That is, some object that has been attached or so to say attested to another, which may be a simple message that has to be transferred over a network between two parties or a file that needs a certification of validity, undeniably associating it with the signer. The signature must have three properties.

First, it must be unique; the signatures of different parties must be different. Second, the signature must not be forgeable; Alice cannot create Bob's signature. Third, the digital signature needs to be verifiable; so anyone can confirm the authenticity.

We demonstrate the concept of digital signatures with a simple example. Suppose Alice wishes to sign a message m, which is ordinarily a hexadecimal string, with public key e, private key d, and public modulus of N, she can do it so by encrypting with her private key and generating the signature as

$$S = m^d \pmod N$$

Thereafter, any passive verifier, say Victor, can check that S is indeed Alice's signature by decrypting with her public key

$$m = S^e \pmod N = (m^d)^e \pmod N = m^{de} \pmod N = m \pmod N$$

2.4.5 One way Hash Function

A one-way hash is a mathematical function. We say h is the hash of M for hashing function H

$$h = H(M).$$

The electronic equivalent of the document and fingerprint pair is the message and digests pair. To preserve the integrity of a message, the message is passed through an algorithm called **cryptographic hash function** .The function creates a compressed image of the message that can be used like a fingerprint. To check the integrity of the message, we again run the cryptographic hash function and compare the new message digest with the previous one. If both are the same, we can concur that the original message has not been tampered with.

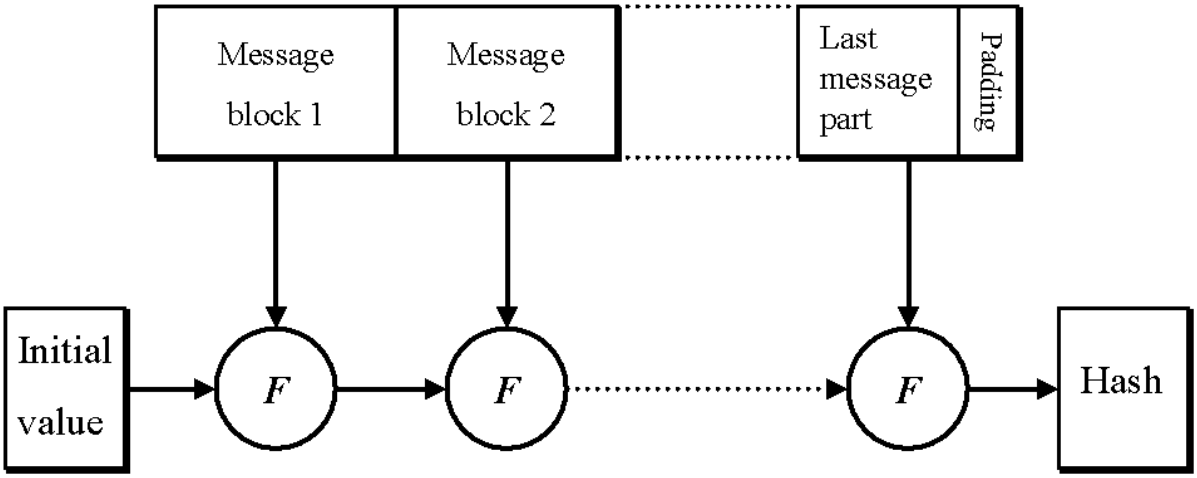


Fig 2.1 A general cryptographic hash function

A one way hash should conform to the following properties[10]:-

- Preimage Resistance – This property concurs that given a hash function H and h = H (M), it should be extremely difficult for any middleman Oscar to find any message M′ such that h = H (M′).
- Second preimage Resistance – This property concurs that a message cannot be easily forged. Given a specific message and its digest, it is computationally impossible to create another message with the same digest.
- Collision Resistance – This criterion ensures that Oscar cannot find two messages that hash to the same digest. Mathematically,

  it is hard to find two messages m, m′ such that H(m) = H(m′).

  A number of hashing algorithms have been proposed in the past, each of which tries to fulfill the aforesaid properties. The most popular and widely used hashing algorithms are enlisted as follows:-

  Message Digest – In regard to message digest or MD, several hashing algorithms were proposed by Ronald Rivest[2], namely **MD2, MD4, MD5**.The MD5 algorithm breaks the message into blocks of 512 bits creating a 128-bit digest. However, MD5 has been broken; an attack against it was used to break SSL in 2008[10].

  SHA (Secure Hash Algorithm)[10] – This is a standard developed by the NSA and published as a Federal Information Processing standard (FIP 180). It is mostly based on the MD5 algorithm. The SHA consists of different versions like the **SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512**. Considering the SHA-512, the newest

version of SHA algorithms, it creates a digest of 512 bits from a multiple block message. Each block is a1024 bits in length. The digest is initialized to a predetermined value of 512 bits. The algorithm mixes this initial value with the first block of message to create the first intermediate message digest of 512 bits. It is the iteratively mixed with the second block to create the second intermediate digest. Finally the $(N-1)^{th}$ digest is mixed with the N^{th} block to create the Nth digest. After the processing of the last block, the resulting digest is the message digest for the entire message. SHA-512 presumes that length of original message is less than 2^{128} bits and creates a 512-bit message digest for the given message.

Apart from these two, other popular hashing algorithms are RIPEMD-128 and RIPEMD-160, GOST, Whirlpool, etc.

2.4.6 Blind Signatures and Blinding

In cryptography, a blind signature, as introduced by David Chaum [1], is a form of digital signature in which the content of a message is disguised (blinded) before it is signed. The resulting blind signature can be publicly verified against the original, unblinded message in the manner of a regular digital signature.

Blind signatures can be used to provide *unlinkability*, which prevents the signer from linking the blinded message it signs to a later un-blinded version that it may be called upon to verify. In this case, the signer's response is first "un-blinded" prior to verification in such a way that the signature remains valid for the un-blinded message. This can be useful in schemes where anonymity is required.

This property of blind signatures is exploited to be used in case of e-voting system where anonymity is a major concern.

There are three parties in the scheme.

– Bob is the signer who has agreed to sign documents blindly.

– Alice is the holder of the message he wants Bob to sign.

– Victor is our verifier who checks whether the signature is Bob’s.

There are five phases defined in the scheme.

- **Key Generation** – Bob sets up the signature by generating all public and secret elements. Public elements are published via a trusted authority while secret elements are kept private.
- **Blinding** – Alice chooses a random elements and masks his message and send the blind message to Bob.
- **Signing** – Bob takes the blind message and signs it. The signature is sent to Alice.
- **Unblinding** – Alice takes the signed blind message, removes the mask (random element) and creates a valid signature for the message.
- **Verification** – everybody who knows the public key, the message and its signature can verify if they match.

If the signer has RSA public key (n, e) and the corresponding private key d, he can sign a message m, $m \in Z_n$ as $s = m^d \bmod N$. Given the signature s of the message m, anyone can verify its validity by checking whether $m = s^e \bmod N$.

The blind version uses a random value r, such that r is relatively prime to N (i.e. gcd(r, N) = 1). *r* is raised to the public exponent e modulo N, and the resulting value $r^e \bmod N$ is used as a blinding factor. The author of the message computes the product of the message and blinding factor, i.e.

$$m' = mr^e \pmod N$$

It then sends the resulting value m' to the signing authority. Because r is a random value and the mapping $r \rightarrow r^e \pmod N$ is a permutation it follows that $r^e \bmod N$ is random too. This implies that m' does not leak any information about m. The signing authority then calculates the blinded signature s' as:

$$s' = (m')^d \bmod N$$

s′ is sent back to the author of the message, who can then remove the blinding factor to reveal *s*, the valid RSA signature of *m*:

$$s = s'.r^{-1} \pmod N$$

This works because RSA keys satisfy the equation $r^{ed} = r \pmod N$ and thus

$$s = s'.r^{-1} = (m')^d\, r^{-1} = m^d\, r^{ed}\, r^{-1} = m^d\, r\, r^{-1} = m^d \pmod N$$

and hence s is the signature of m.

2.5 Various Approaches

This section gives a brief introduction to the approaches used in various voting schemes. While they do not accurately give a picture of the exact procedures followed, the approaches can be seen as the basic framework of different voting schemes

For any voting scheme, privacy seems to be the most important issue. Up to now, only a few approaches to achieve privacy have been invented. Privacy means that the link between the voter and his vote is disposed or inaccessible to everyone (including authority), even if all of the public communication is monitored.

Privacy can be accomplished in the following ways:-

- It is easy to view the vote, but impossible to trace it back to the voter.
- Simultaneous determination of the voter and the vote is impossible.
- While it may be difficult to see the vote, identity of the voter is traceable.

From the above, the first and second approaches have to use untraceable anonymous channel for casting the votes.

2.5.1 Schemes Based on Blind Signatures

Blind signature schemes exist for many public key signing protocols. The message to be signed is designated the value *m*, which is considered to be some legitimate input to the signature function. As an analogy[9], consider that Alice has a letter which should be signed by an authority (say Bob), but Alice does not want to reveal the content of the letter to Bob. She can place the letter in an envelope lined with carbon paper and send it to Bob. Bob will sign the outside of the carbon envelope without opening it and then send it back to Alice. Alice can then open it to find the letter signed by Bob, but without Bob having seen its contents.

More formally a blind signature scheme is a cryptographic protocol that involves two parties, a user Alice that wants to obtain signatures on her messages, and a signer Bob that is in possession of his secret signing key. At the end of the protocol Alice obtains a signature on *m* without Bob learning anything about the message.

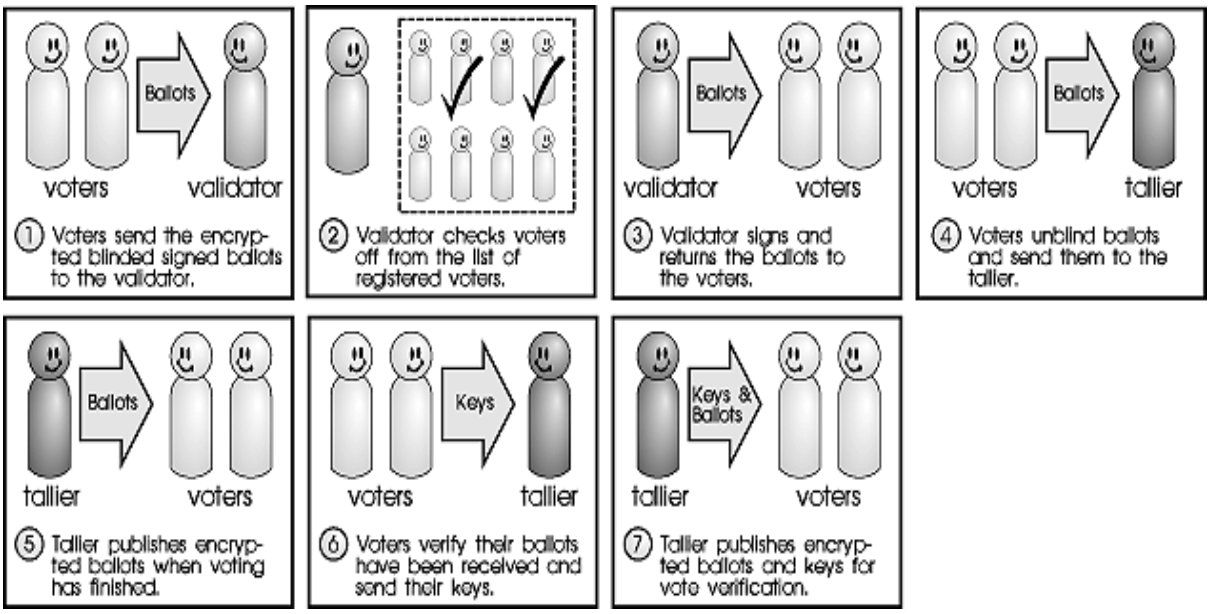


Fig 2.2 Application of Blind Signatures to voting

In the voting phase, the voter sends a ballot containing the token, which basically consists of some data like the voter id, random numbers and his vote through the anonymous channel to the authority. The authority will not accept the ballot with invalid token or with the token that has already been used. This ensures that only eligible voters can vote and that they can vote at most once. As no one can make any connection between the voter and the token or trace the casted ballot back to the voter, no one can deduce anything about how the voter voted. Hence, the privacy is achieved.

A number of protocols exist on the basis of blind signature schemes and a few of them are discussed in the sections to follow.

2.5.2 Schemes Based on Homomorphic Encryption

Homomorphic encryption[16] is a form of encryption where a specific algebraic operation is performed on the plaintext and a possibly different algebraic operation is performed on the ciphertext. The major drawback of these schemes is that they do not support write-in votes.

On the other hand, these schemes perform considerably faster than other types, mostly due to the speed in the tallying phase.

The principal guiding factor in this scheme is the homomorphic property:-

$E(m_1) + E(m_2) = E(m_1 + m_2)$

where E represents the encryption done on messages m_1 and m_2. The $E(m_1) + E(m_2)$ is a calculation in a group G, whereas $E(m_1 + m_2)$ is a calculation in a group H. The `+' is a group operator corresponding to each group, and may be different for G and H.

Advantage of the homomorphic property is the votes can be counted and verified without knowledge of the individual votes. After the election, the encrypted votes are combined into a single, encrypted, quantity. The authorities then decrypt this tally, in the group H. Due to the homomorphic property, this quantity should equal the quantity resulting from the decryption of each of the individual votes in group G. In this way, tallying is done without learning the individual values of the votes. Thus, anonymity is maintained.

Chapter 3

Existing Voting Schemes and Algorithms

In spite of the presence of numerous cryptographic techniques and schemes, a completely practical or theoretical solution for internet voting has never been laid down. A number of practical voting schemes have been proposed, with widely differing security properties. In the sections to follow, we investigate the working of a few protocols which are popularly followed as a base, the schemes introducing new ideas and the schemes efficient in practice.
Initially protocols basing on anonymous channels were proposed for casting the ballots. Later, the schemes exploiting homomorphic encryption were introduced. The property of receipt-freeness was introduced to be satisfied, so schemes were devised that assured of receipt-freeness. Schemes that use blind signatures are quite popular simply because the fact that they are highly efficient and care conducive for any kind of voting scheme. Schemes using blind signatures suffer from the lack of universal verifiability in most cases. This is overcome by the schemes using homomorphic encryption but the computation and communication complexity becomes a serious overhead. As also, they cannot be used for any kind of voting unlike the blind signatures schemes.

3.1 Homomorphic Encryption Models

A number of protocols have been proposed which conform to methods of homomorphic encryption. The first scheme using homomorphic encryption had been proposed by Benaloh and Yung[18]. Further modification to this model was carried out by Sako and Kilian[19] to improve communication efficiency. Thereafter the model proposed by Cramer, Gennaro and Schoenmakers[8] which was a relatively simple and efficient scheme. Benaloh and Tuinstra[17] introduced concept of receipt freeness which was later disproved.
In this model, the voter sends his encrypted vote through a public channel. The vote can be decrypted by any set of at least “t + 1” authorities, and any set of the “t” authorities cannot decrypt the encrypted vote.
This model can be implemented in two ways:

• A key to decrypt the vote is shared between any set of "t + 1" authorities which is known as threshold public-key cryptosystem, as in ElGamal cryptosystem.

• Each authority has its own instance of the cryptosystem. The voter shares his encrypted vote among the N authorities using (t+1, N) secret sharing scheme .The voter sends to the each authority its encrypted share.

This will prevent malicious authorities to abuse their role and to violate voter's privacy.

Encryption method used for encrypting votes is homomorphic, i.e. Multiplication of the encrypted votes v1,v2: E(v1) ⊗E(v2) is an encrypted sum of the votes E(v1⊕ v2).

In a yes/no voting, votes are represented by +1 for yes and -1 for no. Let be p and q be large primes such that q is a factor of p-1 and let g Є Z_p be an element of order q. The secret encryption key is x Є Z_q and the public encryption key is

y = g^x mod p, and w=y^k g^v modp ,

where k is a random number in Z_p .(Z,p) is decrypted by taking w/Z^x modp and by comparing the result with g modp and g^-1 modp .Each voter encrypts his/her vote with the public encryption key of a voting authority and then publishes the encryption on a bulletin board, together with a proof of correctness: that the encryption contains a valid vote

At the end of the voting period the authorities "multiply" all the received encryptions to get an encryption of the tally. The authorities then jointly decrypt this. The final tally can be checked for accuracy by all parties. So we are assured of universal verifiability. For robustness the encryption procedure is distributed among n authorities using threshold cryptography.

An election system based on the Cramer et al scheme [8] has been implemented and piloted on a limited basis. A drawback of such schemes is their reduced flexibility, as the votes are essentially limited to yes/no value. In addition, the Cramer et al scheme which uses ElGamal encryption has a relatively high computational complexity, if the number of candidates is large.

Alternative homomorphic encryption voting schemes have been proposed for which the computational complexity is either linear, or even logarithmic.

3.1.1 Cramer et al Protocol

Cramer et al protocol[8] has the voter sharing his vote among the authorities using secret sharing scheme. This protocol uses the ElGamal cryptosystem.

A simple yes-no protocol proposed by Cramer is suggested here[14].

In the initialization stage, the authorities share the decryption key s. Public key (p, g, h), commitments of the shares $h^j = g^{sj}$ and a fixed generator G of G_q are published.

In the voting stage, the voter V_i chooses his vote: $m_0 = G$ for yes-vote, $m_1 = 1/G$ for no-vote. The encrypted vote is of the form $(x, y) = (gk, h^k m_b)$, where k is random and $b \in \{0, 1\}$. Voter adds a proof that his vote is of the correct form.For this, a non-interactive proof that $\log_g x = \log_h(y/G)$ V $\log_g x = \log_h(yG)$ is used. The encrypted vote along with the proof of validity is sent across to bulletin board.

In the counting phase, the validity proofs are checked and the product of all valid encrypted votes is calculated. The authorities jointly execute the decryption protocol to obtain the value of $W = Y/X^s$. We get W, as per the equation,

$W = G^T$, where T is the difference between the yes and no votes.

This protocol can be extended for a single choice out of many options voting scheme.

Characteristics of the Scheme –

- Privacy - Privacy of the votes is guaranteed partly by the security of ElGamal cryptosystem. Individual vote is hidden for any set of at most t authorities.
- Verifiability - Any passive observer can check the proofs of validity of the ballots, and make a product of the valid votes or check the accuracy of the decryption by checking the proofs of authorities of using correct shares.
- Receipt-Freeness - Voter can reveal to any third party how he has voted by showing randomness k used in the ElGamal encryption. Therefore, this scheme is not receipt-free and prone to coercion.
- Eligibility - Erroneous ballots of forged voters will not pass through the proof of validity. The scheme is resistant up to t malicious authorities.

3.2 Schemes based on mixed-nets

The initial schemes based on mixed nets were devised by David Chaum[15].

The mix-net model is composed of several linked servers where each server accepts a batch of encrypted votes randomizes it and then outputs a batch of permuted votes such that the input is unlinkable with the output vote.

First the authority takes the batch of encrypted votes, permutes it in a random order, and then re-encrypts each encrypted vote. The permutation is known only to the voter.

The permutated batch of re-encrypted votes is published and handed to the next authority; unless the permutation is unveiled to a person no one can map the original vote to the new permuted vote.

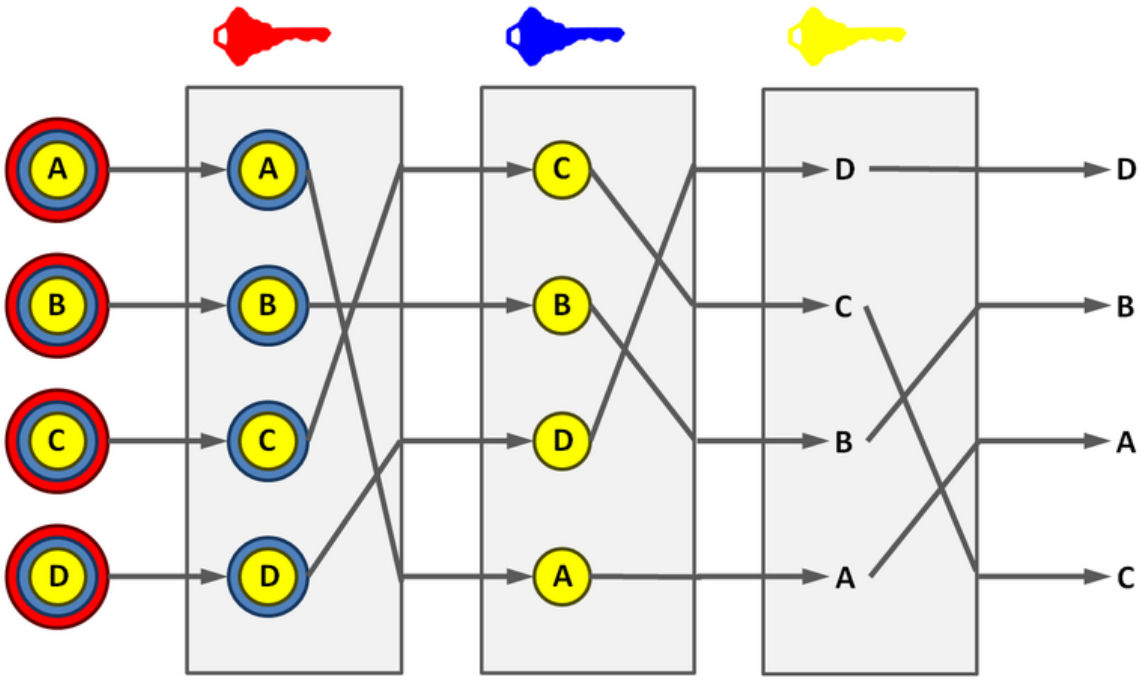


Fig 3.1 A general Mix-net model

The next authority shuffles the votes in the same way as the first authority shuffled the original batch: it permutes the batch in a random order, re-encrypts each vote, and unveils the permutation to the voter publishes the produced batch of votes.

This process is repeated for several times, in the final stage the last authority performs the same process and publishes the final list of permuted and re-encrypted votes.

Therefore, only he knows the voter can map his vote in the final list of the permuted votes.

In large-scale elections, this model of mixed nets is useful because of their universal verifiability, anonymity property.

3.3 Blind Signature Schemes

Our implementation mainly focuses on exploiting the security of schemes based on blind signatures. All schemes on blind signatures follow a basic set of framework: The voter first votes and sends a token along with his encrypted vote. The token can be anything like a valid issued ID signed by the authority as authentic single time only. Then it is blinded and the authority finally unblinds the signature and the counter centre maintains the result.

We move further by examining certain protocols that have been suggested in this area that create a base for the implementation of the internet voting scheme.

3.3.1 The FOO Protocol

The FOO Protocol, as it is popularly known, is an acronym for Fujioka - Okamoto-Ohta Protocol[1]. It was the first protocol to ensure both the privacy and the fairness feature that is so vital to internet voting. The scheme consists of voters, administrator and a counter server with an assumption that communication is done over an anonymous channel. It also requires a bit commitment scheme, a digital signature scheme and a blind signature scheme.

Fujioka, et al. requires the voter to perform three steps.

1. Request the administrator to sign the vote and send it to the counter.

2. Check that the vote is listed by the counter, confirm any of signatures listed, and, if everything is okay, send the keys to uncommit.

3. Confirm that all votes were uncommitted and counted correctly.

We now outline the steps in the FOO protocol as suggested by Fujioka et al.

Initialization phase- The voter selects the vote v_i and completes the ballot ξ (v_i,k_i) using a randomly generated key k_i. Then it encrypts the ballot with an encryption function to generate $e_i=\chi$ (x_i,r_i). Vi the voter signs $s_i=\sigma$ (e_i) to e_i and sends <ID_i,e_i,s_i> to the administrator.

Administration phase – The admin checks if the voter is valid, or he has right to vote, if not, his candidature is rejected. So also, an already registered user cannot register again. If the signature s_i received on the message e_i is valid, the admin signs $d_i = \sigma_A$ (e_i) and sends d_i as admin A's signature to V_i. At the end of this stage, the admin announces the valid list of V_i publishing a list containing <ID_i,e_i,s_i>.

Voting Phase – The voter realizes the signature of the ballot x_i as $y_i = \sigma$ (d_i,r_i) and checks that y_i is the admin's signature .Vi sends <x_i,y_i> to the counter.

Collection phase – The counter checks the signature yi of the ballot xi using the verification key. If checks returns true, the ballot is updated to a list where the votes are counted and result declared.

In these, the implicit assumption is that the channel used is anonymous and the three parties do not collude with each other.

IMPORTANT PROPERTIES ADDRESSED –

1. Security – Because they are randomly generated and used for a single communication between two parties, no session key is intentionally used more than once by party involved.
2. Privacy – Privacy is maintained by phasing the process and the parties involved, only way of breaking it would be if one of the parties colludes with the other, which according to the assumption is not possible.
3. Unreusability – No voter can vote twice as has been demonstrated in the administration phase.
4. Eligibility – Unregistered voters cannot vote since voting is open only to registered voters and breaking the blind signature scheme is difficult.
5. Fairness – Counting of the ballot doesn't affect the voting process since that stage follows after the voting stage and votes are hidden by a bit commitment scheme.

3.3.2 Randomization enhanced Chaum's scheme

David Chaum was a pioneer in the digital signature based voting scheme. The first proposed Chaum's blind signature scheme[5] somewhat followed the FOO protocol but was a significant improvement over it. But later Coron-Naccache-Stern[6] proposed a signature forgery strategy of the RSA digital signature scheme. The attack is valid on Chaum's blind signature scheme. So instead of following the original blind signature scheme, our implementation follows a method to inject a randomizing factor into a message when it is signed by the signer in Chaum's blind signature scheme[4] such that attackers cannot obtain the signer's signatures of the special form for the attack. Users cannot eliminate these randomizing factors embedded in the signatures obtained from the signer.

The phases and the inherent steps followed in the scheme are described as follows:-

Registration phase:

1. Each voter V_i (V_1 ..... V_n) willing to vote must register himself at certification centre, which provides a unique voter ID_i along with a voting ticket to each legitimate voter V_i .

2. The Authentication center publishes its public data e ,n and a one way hash function such as SHA-1 or MD5.

3. Blinding: Each voter V_i randomly chooses an integer $r_i \in Zn^*$, which is the set of all positive integers less than and relatively prime to n. And also chooses a positive integer u_i less than n. Then it computes

$$\alpha = r_i^e \; H(m_i)(u_i^2 + 1) \bmod n.$$

And then sends α to the Authentication Center. After receiving α, the Authentication Center randomly selects a positive integer x_i, the voter V_i chooses an integer $b_i \in Zn^*$ and computes

$$\beta = b_i^e (u_i - x_i) \bmod n.$$

Finally the voter V_i submits (β , ID_i) to the Authentication Center.

4. Signing: this process is done by the authentication center. After receiving β , it computes $t_i = (\alpha (x_i^2 + 1) \beta^{-2})^d \bmod n$. The integer x_i is called the randomization factor. Then it sends t_i to the voter V_i.

5. Unblinding: this process is done by each legitimate voter V_i. After receiving t_i , the voter V_i computes

$$C_i = (u_i\, x_i + 1)\, (u_i - x_i)^{-1} \bmod n$$

$$S_i = r_i^{-1}\, b_i^2\, t_i \bmod n$$

S_i is the signature of the authentication center on message m_i. To verify the authenticity of the signature, he/she examines if the following equation holds good

$$S_i = H(m_i)\, (C_i^2 + 1) \bmod n.$$

When the deadline of registration is over, the authentication center displays (ID_i , S_i , C_i)in to a list. It publishes the list after the election date

Voting phase:

1. Each voter V_i retrieves the signature of the authentication center and checks its validity. It checks that the voter has not previously casted any vote. Then he sends (ID_i , S_i , C_i) to the Counting center.

2. The Counting center verifies the signature of the ballot. If verified, it puts (ID_i , S_i , C_i) in to the list . It publishes the list after the election date.

Counting phase :

1. Each voter V_i verifies whether S_i , C_i are in the list. If not, then he can complain by showing the valid pair.

2. After the deadline of confirmation is over, counting is done. Then it publishes the final result of election.

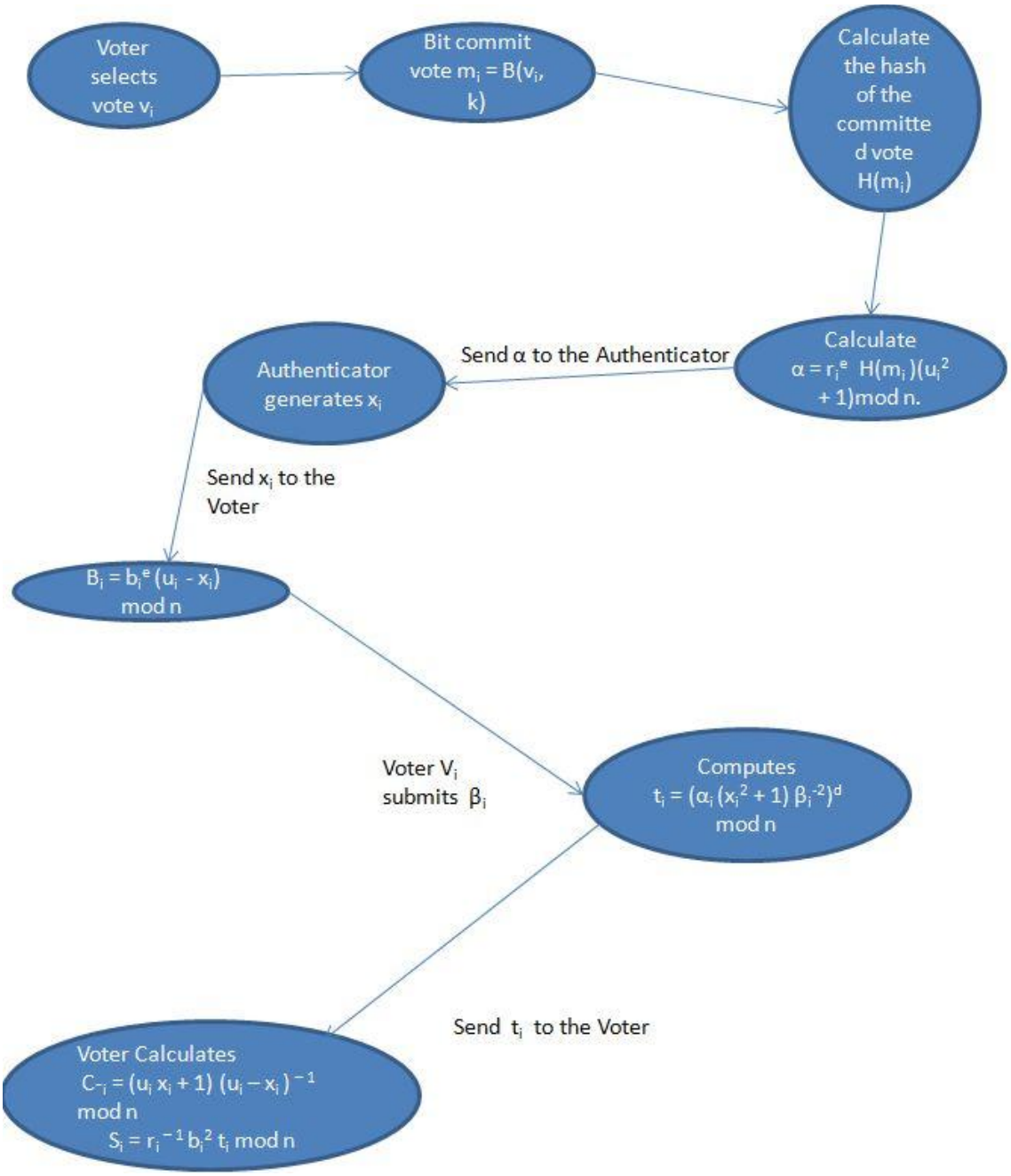


Fig 3.2 Randomization enhanced Chaum's scheme

3.1 Comparison of the various protocols studied

The Fujioka et. al. protocol is considered to be one of the most suitable and promising for large-scale elections, since the communication and computation overhead is fairly small even if the number of voters is large. Moreover, this type of scheme naturally can allow multiple values voting, and is also very compatible with the framework of existing physical voting systems.

Given below is a comparison of various protocols[12] and the properties they satisfy and why a randomized enhanced Chaum's blind signature model based on the FOO protocol has been chosen as the implementation scheme. Evaluating the three different protocols on their feasibility is debatable since requirements of each have to be met. Besides, there have already been mentioned a number of actual implementations based on these protocols that meet some requirements and are practical for large-scale elections. Nevertheless, some of the prerequisites defined in the theoretical analysis of these protocols can be unrealistic to achieve in practice.

Properties	**Fujioka et. al. (based on blind signature)**	**Cramer et. al. (based on homomorphic cryptosystem)**	**Chaum (based on mix-net)**
Privacy	yes	yes	yes
Verifiability	yes	yes	Depends on mix net
Availability	no	yes	no
Integrity	yes	yes	yes
Reliability	yes	yes	Depends on mix net
Incoercibility	no	no	no

Table showing the comparison of various types of voting protocols

Evaluating the three different protocols on their feasibility is debatable since requirements of each have to be met. Besides, there have already been mentioned a number of actual implementations based on these protocols that meet some requirements and are practical for large-scale elections. Nevertheless, some of the prerequisites defined in the theoretical analysis of these protocols can be unrealistic to achieve in practice.

Besides, when comparing the efficiency of voting schemes, one needs to refer to some "reasonable" parameter values. The most important parameters are the number of voters N and the number of options of multiple option question L. Other parameters are the number of authorities M and the trust threshold t. So the suitability and usability of each protocol varies and depends a lot on these actual numbers.

Chapter 4

Implementation

4.1 Required Components

For the implementation of the Secure Internet Voting Protocol, we have chosen JAVA as our base language.

We have used J2SE JDK and NetBeans IDE 6.7

The required database, i.e, the Candidate's database, the eligible Voter's database and the result database were created and maintained by using Java Derby. Derby is based on the Java, JDBC, and SQL standards.

Cryptography in java requires The Java™ Cryptography Extension (JCE). JCE provides a framework and implementations for encryption, key generation, key agreement, Message Authentication Code (MAC) algorithms, etc. It supports symmetric, asymmetric, block, and stream ciphers encryption techniques.

The JCE API covers RSA (Asymmetric encryption) which is used for public key management in the voting protocol. An implementation of the MD5 and SHA1, SHA-256, SHA-512 keyed-hashing algorithms is also covered in the JCE API which is used in the protocol. This framework includes everything in the javax.crypto package.

Additionally, different packages or jar files were also used in the implementation of the protocol. By using Java Derby as the database management system, we used derby.jar file and in addition to it for the implementation of decryption we have used the freely available gnu.crypto jar file.

4.2 Implementation Steps

Architecture:

The architecture of our system mainly consists of multiple voters, a authenticator server and a counting server. The voter blinds his vote and sends it to the authentication server which is then signed by the authenticator's signature and sent to the voter. The voter sends his signed vote to the counting server through anonymous channel. Java provides basic networking functionality used to connect the voter with the servers. This includes the ability to create

sockets and send object streams through them. Objects are also serializable, that is, transformable to and from byte arrays, which are more readily transferable.

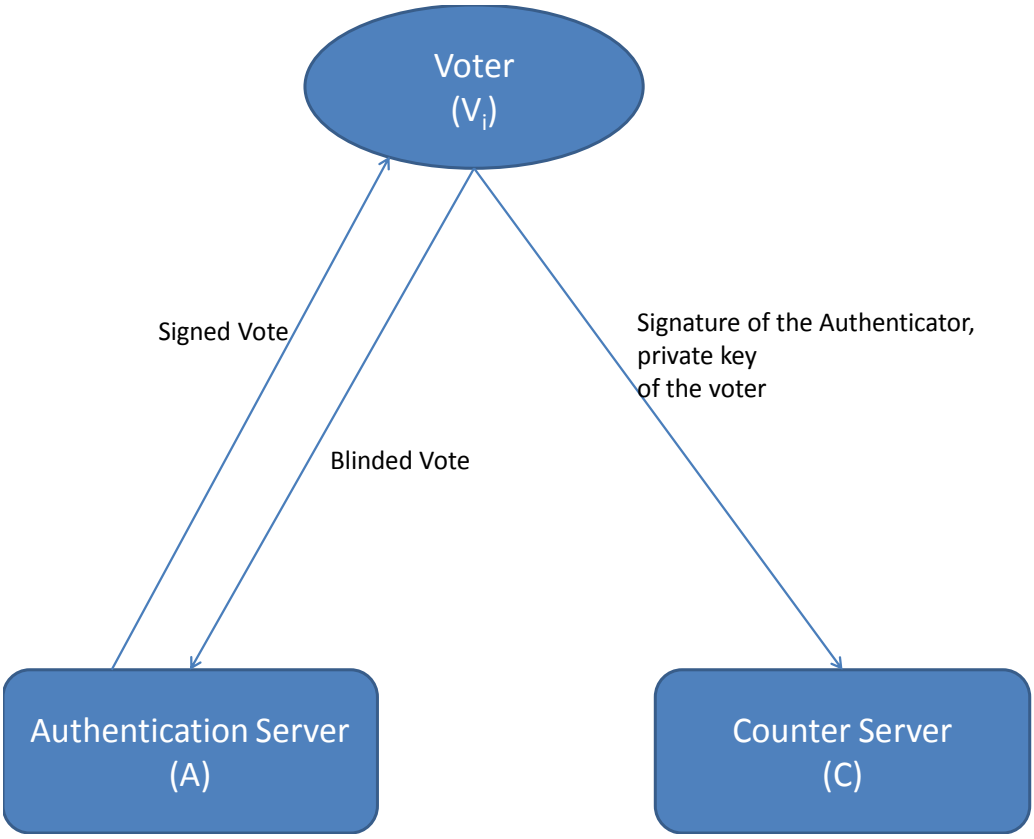


Fig 4.1 Architecture of the voting system

Registration Phase:

First step in the voting system is registration where the voter has to go to a registration office and show his voters ID card and if he is eligible to vote he can register for voting. He has to remember his user name and the password he has used while registration to vote successfully. After the user has registered successfully he is given a randomly generated number which is his ID number and should be noted down and kept secret for the voting process. A voter table is created in this stage where the voter's username, ID, password and his voting status (whether he has voted? Initially marked with 0 for false) are stored.

Login Phase:

When the date for voting arrives the voter can login using his username, password and the unique ID number given to him the registration phase.

The voter's input is verified in the registration file if it is correct and the voter has not voted yet then he is redirected to the voting page. The voting page displays the candidates' names. Each candidate has his candidate ID which is kept secret and known only to the authenticator.

As according to the set of guidelines laid down before the implementation, the protocol satisfies different properties. A voter cannot vote twice. Once he has logged in , his unique ID is generated , which is to be used only once , logging with the same ID again returns an invalid message.

Authentication:

When the voter has successfully logged in he is connected to the authentication server. The server runs a number of threads for multiple connections, which will respond to secure connection request from a voter. The voter selects a vote v_i and bit commit the vote with a randomly selected key ki which is a 512 bit key generated by SecureRandom method.

Then the voter sends the bit committed vote to the server. The server generates SHA-512 hash of the bit committed vote using the java.security.MessageDigest

Package.

MessageDigest md=MessageDigest.getInstance("SHA-512")

The next step is blinding, in this step each voter V_i randomly chooses an integer $r_i \in Zn^*$, which is the set of all positive integers less than and relatively prime to n. And also chooses a positive integer u_i less than n. Then it computes

$\alpha = r_i^e \; H(m_i) (u_i^2 + 1) \bmod n.$

Then sends α to the Authentication server. The Authentication server randomly selects a positive integer x_i , the voter V_i chooses an integer $b \in Zn^*$ and computes

$\beta = b_i^e (u_i - x_i) \bmod n.$

Finally the voter V_i submits (β , ID_i) to the Authentication Center.

The next stage in this process is signing; this process is done by the authentication server. After receiving β, it computes

$t_i = (\alpha_i (x_i^2 + 1) \beta_i^{-2})^d \bmod n$

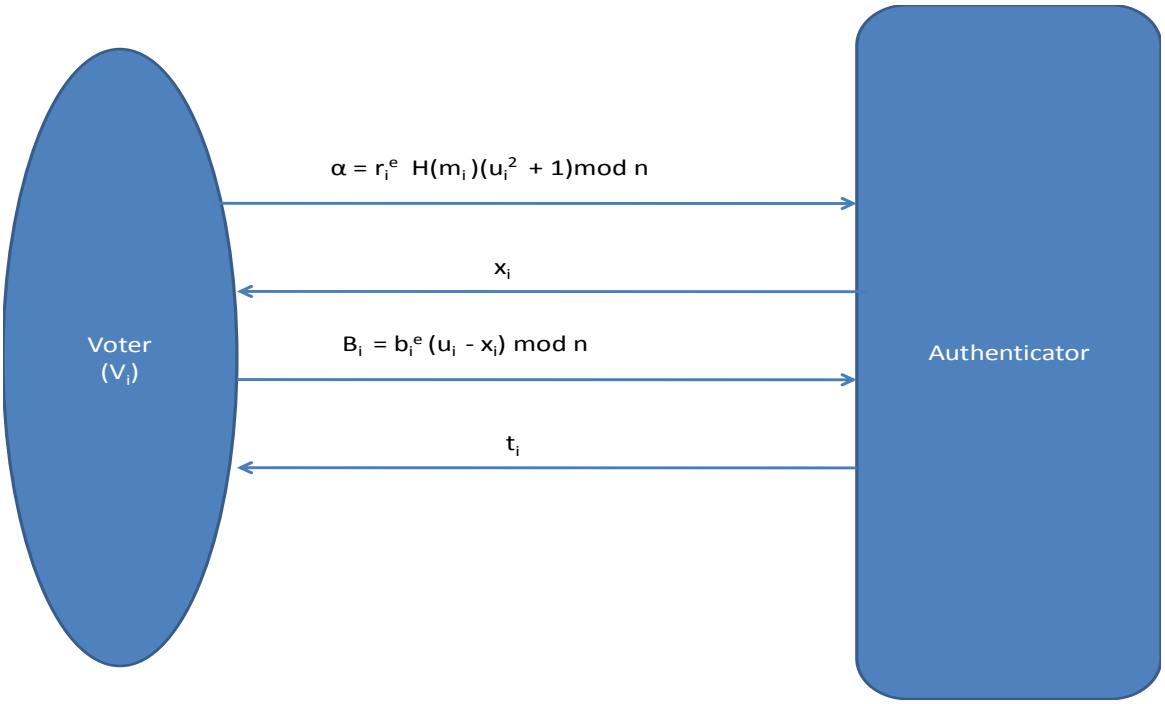


Fig 4.2 Interaction between the voter and authenticator

The integer x_i is called randomization factor. Then it sends t_i to the voter V_i.

After receiving t_i , the voter V_i computes

$$C_i = (u_i\ x_i + 1)\ (u_i - x_i)^{-1} \bmod n$$

$$S_i = r_i^{-1}\ b_i^2\ t_i \bmod n$$

S_i is the signature of the authentication server on message m_i. To verify the authenticity of the signature, he/she examines if the following equation holds good

$$S_i = H(m_i)\ (C_i^2 + 1) \bmod n.$$

Counting Phase:

The Counting center verifies the signature of the ballot. Then the voter sends the key ki used in bit commitment .The counter open the ballot and and it puts (Si , C_i , CID) in the list. After the voting process is over the counter displays a list. The voter can verify his vote corresponding to the S_i and C_i value of the voter.

Finally when the S and C values match, the table that was created for the results is updated as per the candidates who have been voted . Finally after the completion of voting by all voters or end of the voting, the results are analysed. The rows are sorted by decreasing order of the number of votes in favour of the candidate and the results are declared accordingly.

4.3 RESULTS -

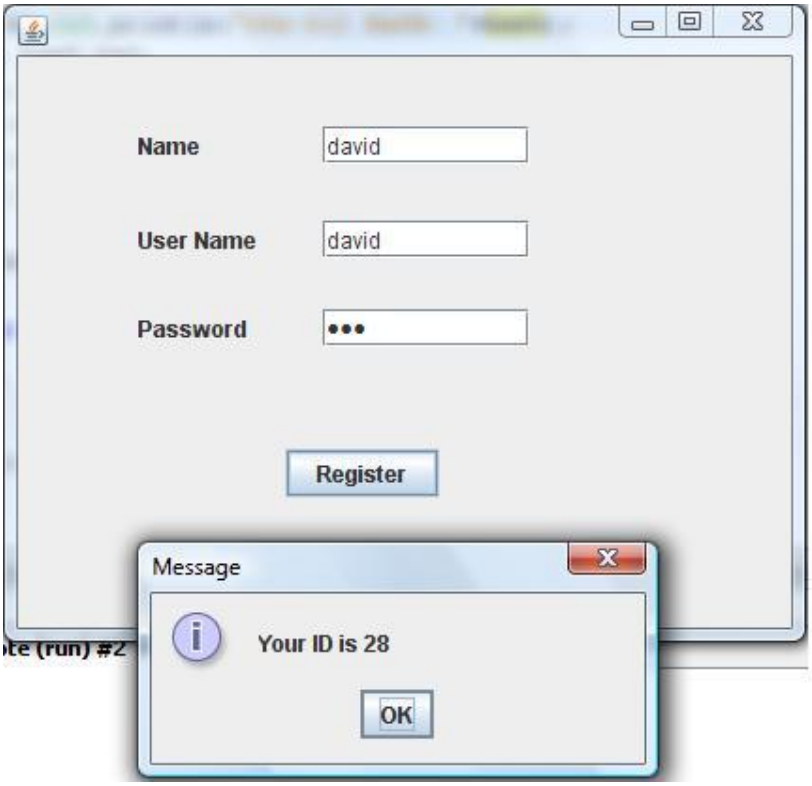


Fig 4.3 Snapshot showing Generation of unique ID

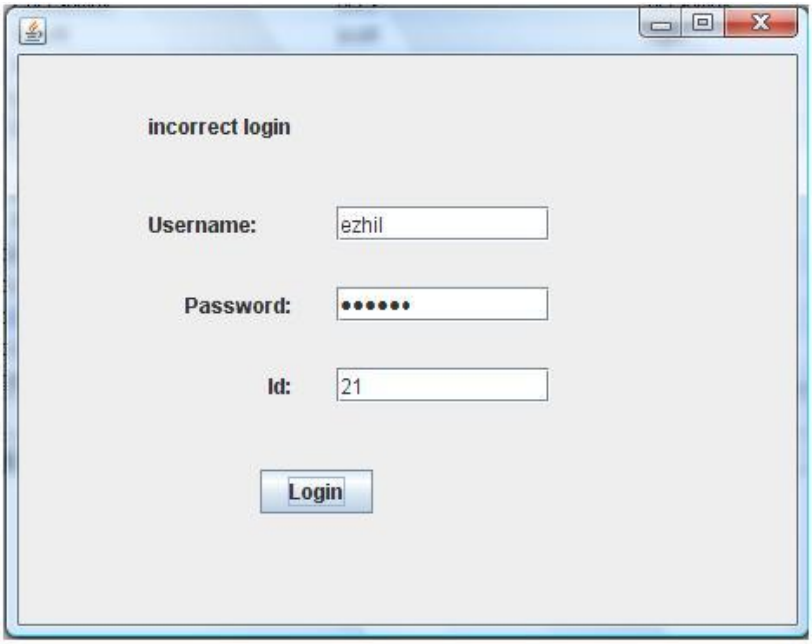


Fig 4.4 Snapshot showing return of "incorrect login" if username, password or ID do not match

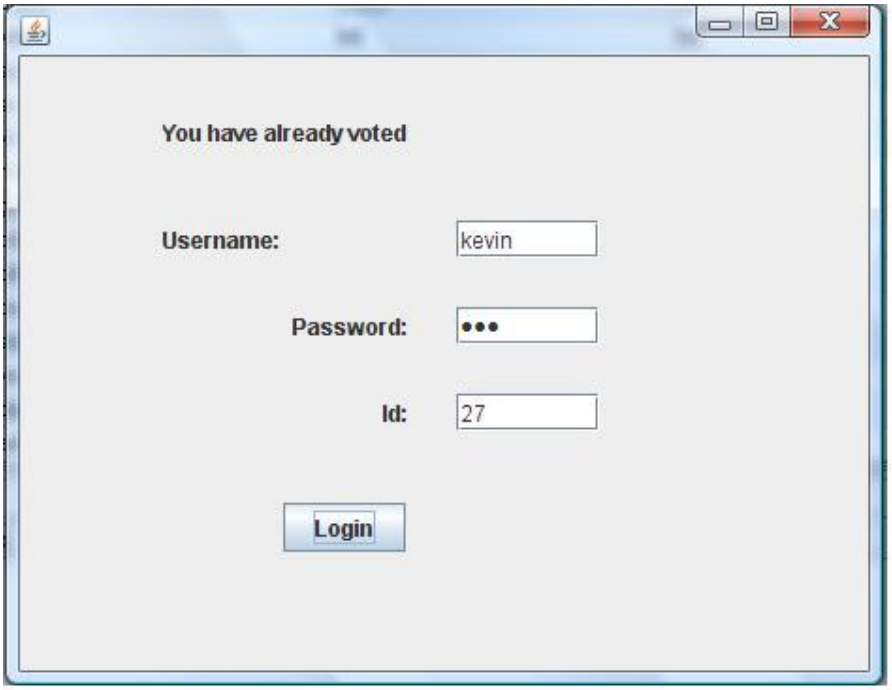


Fig 4.5 Snapshot showing a voter that he has already voted

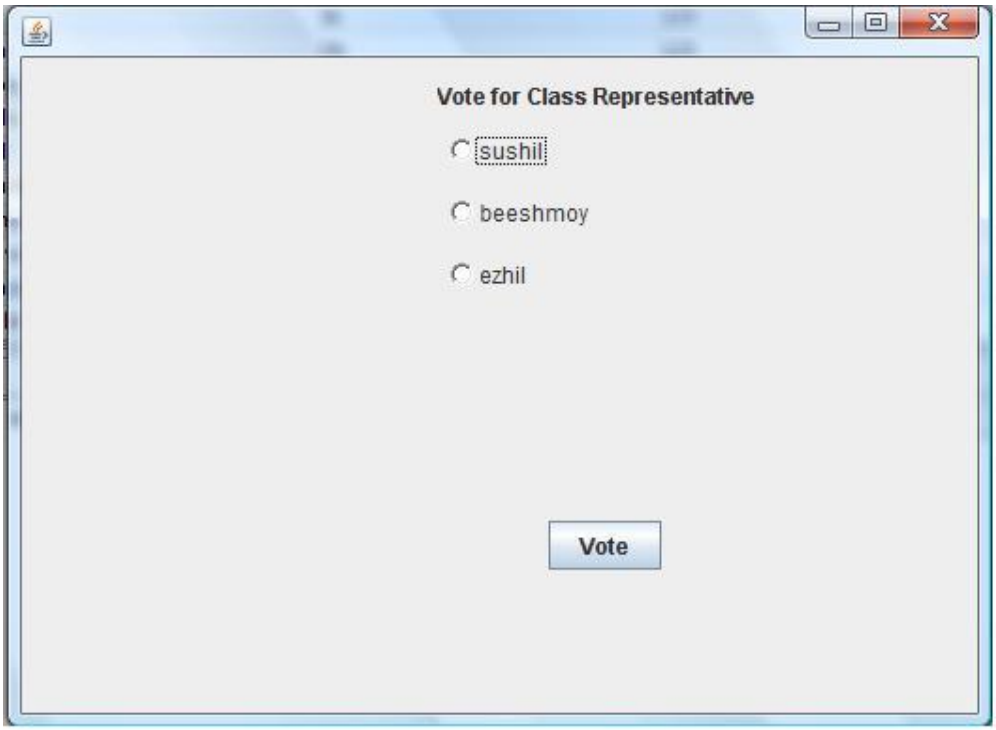


Fig 4.6 The Voting Page snapshot

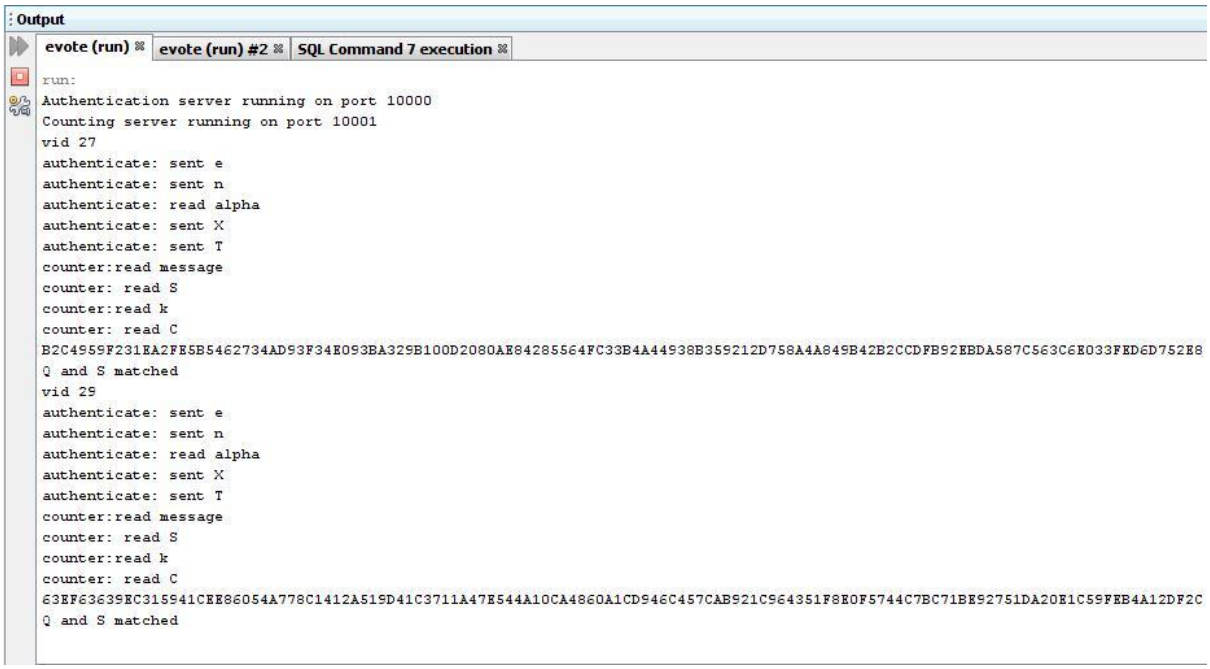


Fig 4.7 Snapshot showing the running of the authentication server and the complete process

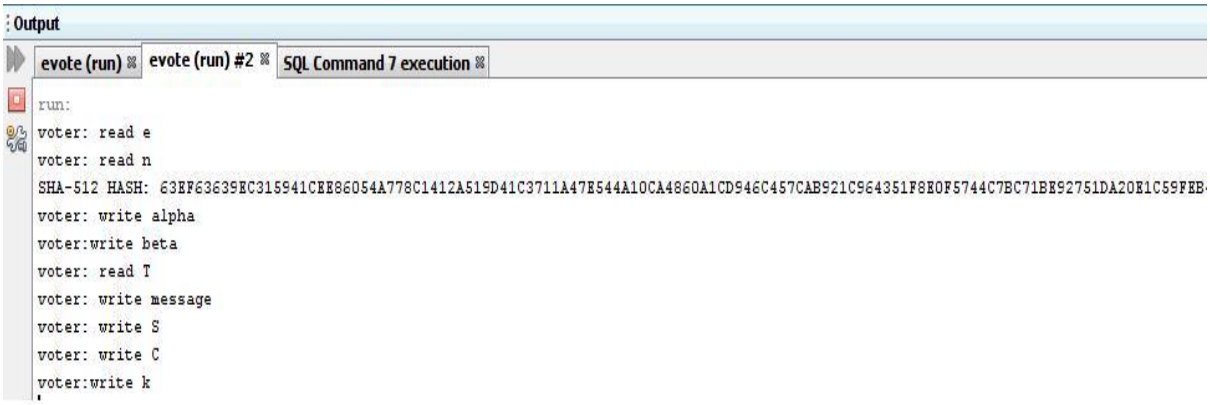
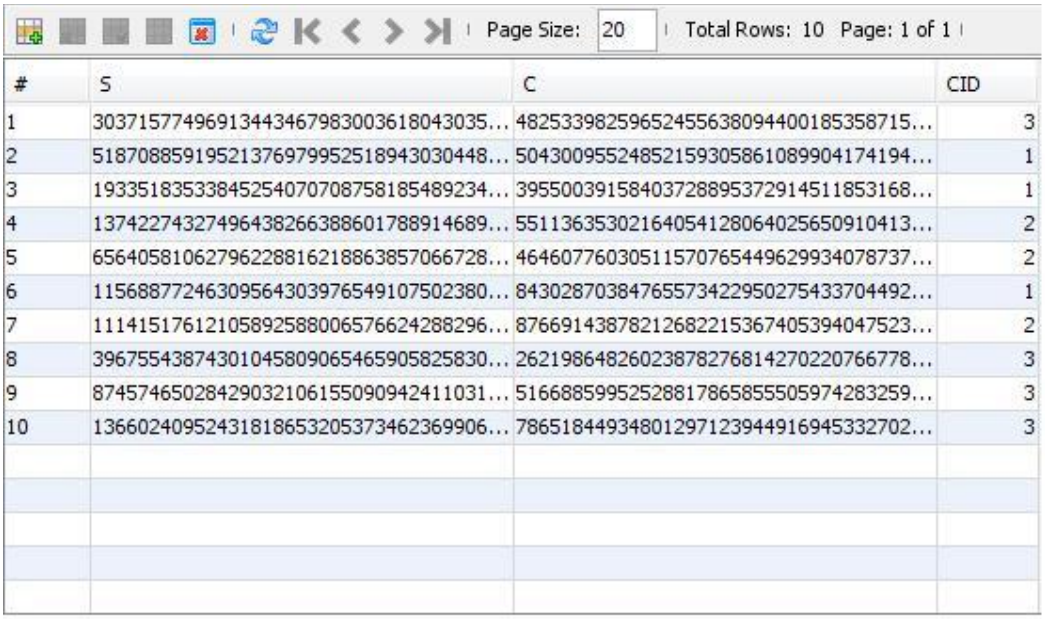


Fig 4.8 Snapshot output of the voter server



Page Size: 20 Total Rows: 10 Page: 1 of 1

#	S	C	CID
1	303715774969134434679830036180430 35...	48253398259652455638094400185358715...	3
2	518708859195213769799525189430304 48...	50430095524852159305861089904174194...	1
3	193351835338452540707087581854892 34...	39550039158403728895372914511853168...	1
4	137422743274964382663886017889146 89...	55113635302164054128064025650910413...	2
5	656405810627962288162188638570667 28...	46460776030511570765449629934078737...	2
6	115688772463095643039765491075023 80...	84302870384765573422950275433704492...	1
7	111415176121058925880065766242882 96...	87669143878212682215367405394047523...	2
8	396755438743010458090654659058258 30...	26219864826023878276814270220766778...	3
9	874574650284290321061550909424110 31...	51668859952528817865855505974283259...	3
10	136602409524318186532053734623699 06...	78651844934801297123944916945332702...	3

Fig 4.9 Snapshot of the result table showing S and C and the candidate id voted against the voter id

Chapter 5

Conclusion

In this thesis, we have implemented a secure internet voting protocol based on randomized enhanced Chaum's blind signature schemes. The blinding factor in each step of the process maintains that breaking this protocol would not be easy. The project work envisages all the core components required for the functioning of the Internet voting Scheme and implementing it under Java makes it quite easy for extensions to various types of polling and voting schemes other than the 1-out-of-L scheme we have implemented. This scheme ensures the privacy of the voters and prevents any disruption by voters or the administrators. The implemented scheme covers most of the security requirements of the internet voting scheme including voting fairness. The problem of computer and the Internet security has taken a prominent and important place in today's research area. Since electronic election is a part of these applications, it is of supreme importance as we will consider its emerging advantages in today's modern life. This problem is open, researches in different universities and laboratories are still going on. Different protocols are emerging by the day, each with a hint of advancement over the other. With the growing use of internet in these days, it is evident that better and more secure protocols would come to the fore and their practicality can be exploited to meet the growing security needs.

REFERENCES:

[1]. A. Fujioka, T. Okamoto, K. Ohta, A practical secret voting scheme for large scale elections, Advances in Cryptology - AUSCRYPT'92, Lecture Notes in Computer Science,718, Springer, pp. 244-251, 1993.

[2]. R. L. Rivest, A. Shamir, L. M. Adleman, A method for obtaining digital signatures and public key cryptosystems, Communications of the ACM, 21(2), pp. 120-126, 1978.

[3] D. Chaum, "Blind signatures for untraceable payment", Advances in cryptology, CRYPTO'82, Lect. Notes Computer Science, (Springer-Verlag,1998), pp. 199-203

[4] Chun-I Fan, W.K. Chen, and Y. S. Yeh, "Randomization enhanced Chaum's blind signature scheme," Computer Communications, vol. 23, pp. 1677–1680, 2000.

[5] D. Chaum, *Blind Signature Systems*, U.S. Patent 4,759,063, 19 Jul 1988.

[6] J.S. Coron, D. Naccache and J.P. Stern, "On the security of RSA padding," Advances in Cryptology – CRYPTO'99, LNCS 1666, pp.1–18, Springer–Verlag, 1999.

[7] A. Menezes, P. van Ooorschot and S. Vanstone, "Handbook of Applied Cryptgraphy," CRC Press, 1997.

[8] R. Cramer, R. Gennaro, B. Schoenmakers. "A secure and optimally efficient multi-authority election scheme" Advances in Cryptology-Eurocrypt 97, LNCS vol. 1233, 103-118, 1997.

[9] Secure Electronic Voting Over the World Wide Web by Mark A. Herschberg Master's thesis, Massachusetts Institute of Technology 1997

[10] William Stalllings, Cryptography & Network Security, Fourth Edition, Pearson Education, 2006

[11] An Efficient Implementation of Electronic Election System by Naznin Fauzia ,Tanima Dey ,Inaba Bhuiyan ,Md. Saidur Rahman.

[12] A Critical View on Internet Voting Technology Eleni Tsekmezoglou, John Iliadis

[13] Tadayoshi Kohno, Adam Stubblefield, Aviel D. Rubin, and Dan S.Wallach. Analysis of an electronic voting system. In IEEE Symposium on Security and Privacy, May 2004.

[14] Zuzana Rjašková , Electronic Voting Schemes Diplomová práca ,April 2002

[15] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM, 24(2):84–88, 1981.

[16] An Analysis of Chaum's Voter-Verifiable Election Scheme , Julie Ann Staub, Master of Science, University of Maryland, College Park, 2005

[17] Josh Cohen Benaloh and Dwight Tuinstra. receipt-free secret-ballot elections (extended abstract). Proc. 26th ACM Symposium on the Theory of Computing (STOCK), 1994.

[18] Josh Cohen Benaloh and Moti Yung. Distributing the power of the governement to enhance the privacy of voters (extended abstract).1986.

[19] Kazue Sako and Joe Kilian. Secure voting using partially compatible homomorphisms. Advances in Cryptology - CRYPTO'94, Springer-Verlag:411–424, 1994