Design and Implementation of Secure Internet Based Voting System with User Anonymity using Identity Based Encryption System

Purushothama B R
Department of Computer Engineering.
National Institute of Technology Karnataka,
Surathkal, INDIA
purus.br@gmail.com

Alwyn R Pais
Department of Computer Engineering.
National Institute of Technology Karnataka,
Surathkal, INDIA
alwyn.pais@gmail.com

Abstract

With Internet becoming ubiquitous, electronic transactions over the Internet have become an integral part of day to day life. The Internet is used for more and more secure transactions like banking, shopping, submitting tax returns etc. In a way, the need for a secure Internet based electronic voting system is an obvious demand. The task of designing a secure Internet based voting system is a cryptographic challenge. This paper proposes and discusses the design and implementation of secure Internet based electronic voting system using Identity Based Encryption System (IBES). This proposed system satisfies various security requirements like, privacy, anonymity, eligibility, accuracy, fairness, uniqueness, verifiability and receipt freeness. Total user anonymity is achieved using IBES.

Key words— IBES, User Anonymity, and Voting.

1. Introduction

As new services like e-commerce, e-cash and e-government using cryptographic primitives become popular over the Internet, the possibility of electronic voting over the internet also attracts great interest.

Election is a fundamental tool that allows people to choose who will govern them. Traditional voting mechanisms demand the voter to come in person to vote. This results in low participation rate of voting. Certain voters residing in sparsely populated areas and who work far away from the voting centers can use vote-by-mail mechanisms to cast their vote. This method is time consuming and cumbersome for authorities to manage, as this involves extra work to send, collect and count the ballots manually. A secure electronic voting through Internet can overcome these problems. By using an Internet based voting system a voter can vote from his home, office or from anywhere.

Internet voting must meet security requirements such as anonymity, privacy, eligibility, authentication, accuracy, completeness, fairness, verifiability, avoiding double voting and receipt freeness. These requirements make Internet voting much more challenging than electronic commerce or electronic government applications.

In this research, we have designed and implemented a prototype of an Internet based voting system that satisfies the security requirements for a safe election and achieves user anonymity using IBES. This is achieved by designing some protocols that guarantee those requirements.

The rest of the paper is organized as follows. Related work is explained in section 2. Section 3 elaborates on the security requirements that an Internet based voting system should satisfy. Security mechanisms used in the Internet voting system are discussed in section 4. Our proposed Internet voting system architecture is given in section 5. Security analysis is done in section 6. Section 7 explains the implementation details. Section 8 gives the performance measurement and theoretical analysis of our model. Section 9 concludes the paper followed by references.

2. Related Work

There is an extensive set of requirements that any e-voting system should satisfy [1]. The formal definitions of security requirements for cryptographic voting protocols such as privacy, eligibility, uniqueness, fairness, receipt freeness, accuracy and individual verifiability are provided in [2]. Rationalized unified process is used to identify the requirements of an adequately secure e-voting

978-0-7695-3811-2/09 $26.00 © 2009 IEEE
DOI 10.1109/SCC.2009.30

system. There are few specific design principles for the secure e-voting system [3]. Voter Privacy which is one of the main security issues in the election process can be guaranteed by using blind signatures for confidentiality and voter's digital signature for voter's authentication [4]. IC card (Integrated Circuit card or smart card) is used for performing a variety of activities like, bank transactions, public telephone calling, remote password authentication, etc. IC card can be used to authenticate the identity of the voter [5].

The electronic voting systems can be implemented based on various cryptographic primitives such as homomorphic encryption, mixnet and blind signature. There are implementations of the electronic voting systems which don't use the above mentioned cryptographic primitives [6].

Various schemes suffer from the key distribution problem which is solved by using Public Key Infrastructure (PKI) [11].

In [7] PKI based system is used to design an Internet based voting system. There is redundant complexity involved in the PKI's certification, certificate chain determination and certificate verification process.

The system developed using the blind signature scheme [8] requires the voter to download and install the client application before using the voting system.

To remove the redundancy associated with the PKI, a new approach called Identity Based Encryption is used in our solution.

The developed Internet based secure voting system using IBES improves over and removes the disadvantages of the systems in [7][8].

3. Security Requirements

This section elaborates the security requirements that an Internet voting system should meet to make it secure. We claim that our system satisfies these requirements.

There is scope for fraud and corruption without these security requirements. To avoid all these problems of security, an Internet based voting system should satisfy the following security requirements.

- **Privacy:** No connection between voter and vote.
- **Eligibility:** Eligible voters can vote.
- **Uniqueness:** Each eligible voter has voted only once. Avoid double voting.
- **Fairness:** Result is not published till the end of the election. Counting comes after the voting stage. No one can guess the content of any cast vote.
- **Authentication:** Only authorized voters should be able to vote.
- **Receipt-freeness:** Voter is not identifiable from the receipt. Vote is not revealed from the receipt. Voter cannot prove his vote. Vote selling/buying is prevented.
- **Verifiability:** Voter can validate that his vote is recorded correctly. Each eligible voter can verify that his vote is counted correctly contacting the administrator after the voting process is complete. Accuracy is taken care of.
- **Anonymity:** Voter identity is not revealed to anyone.
- **Voter selling /trading:** It should not allow the voter to establish to a third party that he/she has voted in some way.

4. Security Mechanisms in the Voting System

In this research, we have developed a secure Internet based voting system using the Identity Based Encryption System. The user anonymity, privacy and confidentiality are provided by the IBES.

Traditional approaches to key management including symmetric key management, PKI, have fallen short in meeting the requirements of an effective enterprise key management system.

Identity Based Encryption(IBE), uniquely meets all requirements for an effective enterprise key management system by encrypting data, authenticating users and decrypting data, jointly managing keys with partners, delivering keys to trusted infrastructure components, recovering keys, and scaling for future growth. IBE meets these requirements in a cost-effective and user-accessible manner, ensuring adoption and ultimately, the security of electronic communications.

4.1 Identity Based Encryption System

Identity Based Encryption System is a public key cryptosystem designed mainly to remove the redundant complexity involved in the Public Key Infrastructure's certification, certificate chain determination and certificate verification process. In this system, a recipient's well known unique *ID*, like National Identity Number, email address, a mobile phone number, an IP address, a URL, etc., is used as the public key for encryption. The system architecture has a trust component, the Private Key Generator (PKG), which ensures that only the owner of this particular unique identity has the private key for this *ID*, and hence none other can decrypt it. Figure 1 depicts the system architecture and the various steps for secure message transfer.

The concept of IBES was proposed by Shamir in 1984. But, the first successful and computationally feasible system was published in 2001 by Dan Boneh and M Franklin [9]. Their System makes use of a concept called Weil pairings, a bilinear function.

The *ID* based scheme consists of four algorithms. Setup, Extraction, Encryption, and Decryption. Setup is run by the PKG to generate a master key and the system parameters. This is done on input of a security parameter *k*, which specifies the bit length of the group order and is regarded as the key size of the ID-based scheme. The extraction algorithm is carried out by the PKG to generate a private key corresponding to the identity of the user. As with standard public key cryptography, the encryption algorithm takes a message and a public key as input to produce a cipher text. Similarly, the decryption algorithm is executed by the owner of the corresponding private key to decrypt the encrypted text. These four functions are described as follows:

Setup: With the parameter *k*, the algorithm works as follows:

a) Generate a random *k*-bit prime *p*, two groups $(G_1, +)$; $(G_2, *)$; of order *p*, and the Weil pairing $e: G_1 \times G_2 \rightarrow G_2$. Choose an arbitrary generator $P \in G_1$.
b) Pick a random number $s \in Z_p^*$ and set $P_{pub} = sP$.
c) Choose cryptographic hash functions $h_1 : \{0,1\}^* \rightarrow G_1^*$ and $h_2 : G_2 \in \{0,1\}^n$ for some *n*. The public system parameters are $p, G_1, G_2, P_{pub}, e, n, P, h_1, h_2$ and the master key *s* is kept secret by the PKG.

Extraction: For a given string $ID \in \{0,1\}^*$ as the public key, the algorithm works as follows.

a) Compute $Q_{ID} = h_1(ID) \in G_1$.
b) Set the private key $K_R = sQ_{ID}$, where *s* is the master key held by PKG.

Encryption: To encrypt a message M under the public key ID, the algorithm works as follows:

a) Compute $Q_{ID} = h_1(ID) \in G_1$.
b) Choose a random $r \in Z_p^*$.
c) Set the cipher text to be,

$C = (U, V) = (rP, M \oplus h_2(e(Q_{ID}, sP)^r))$.

Decryption: To decrypt a cipher $C = (U, V)$ encrypted using the public key ID, the algorithm uses the private key $K_R = sQ_{ID}$ to compute $M = (V \oplus h_2(e(sQ_{ID}, U))$. This decryption procedure yields the correct message due to the bilinearity of the Weil pairing, i.e.,

$$e(sQ_{ID}, U) = e(sQ_{ID}, rP) = e(Q_{ID}, sP)^r.$$

It can be observed that as public keys are derived from identifiers, IBE eliminates the need for a public key distribution infrastructure, which means there is no public key exchange before a message transfer as in other crypto systems. Hence, man-in-the middle attack is not possible. Also, there is no certificate retrieval, nor certificate chain determination and certificate verification as in PKI, which makes the system efficient. The authenticity of the public keys is guaranteed implicitly as long as the transport of the private keys to the corresponding user is kept secure (Authenticity, Integrity, and Confidentiality).

A Symmetric algorithm DES and SHA-1 hash function are implemented and used to provide the security to the trusted voting system entities.

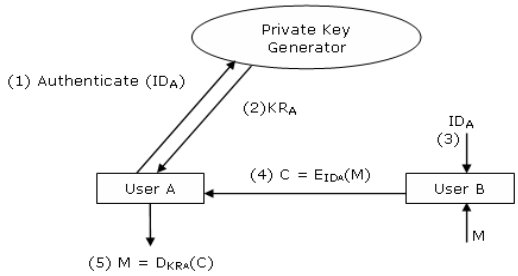

Figure 1. Identity Based Encryption System

5. Internet Based Voting System Architecture

In this section, we describe the system architecture and the voting stages of secured Internet based voting system proposed by us. The system architecture is as depicted in Figure 2. There are 8 entities involved in the system.

- Voter
- Administrator/Registrar
- Validator/Ballot Server
- Counter/Collector/Tallier
- National Registration Database (NRDB)
- E vote database
- Count Database
- PKG Server.

It is assumed that preferably the Private Key Generator (PKG) entity is the entity operating the voting system. All the entities work in coordination with the PKG. PKG is the trusted component. NRDB contains information about all the voters.

There are four phases in the voting process.

- Voter Registration process
- Authentication and Ballot Distribution
- Casting vote

- Collecting, storing and counting the votes.

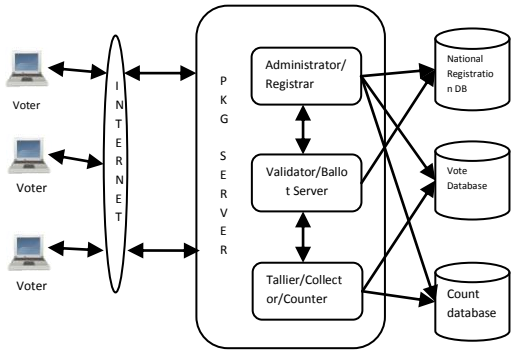

Figure 2. Internet based voting system architecture

To cast a vote a person should be a registered and eligible voter. If eligible, the voter is given a ballot containing the list of candidates contending for the election. The voter selects the candidate of his interest and casts his vote. The vote information is stored in the vote database. After the voting process is complete, the votes are counted and the count of the votes is stored in count database and the result is published.

To maintain the user anonymity, first the user communicates with the PKG Server to get the PseudoID [10]. PseudoID is a mapped point of *ID* on the curve.

5.1 Generating an anonymous PseudoID

It is assumed that the communication happens through Secure Socket Layer (SSL).

- Voter requests for server information
- Voter generates a temporary symmetric key, SK_T.
- Voter sends SK_T and his/her National Identity card number (*ID*) encrypted using server's public key.
- Server extracts the user's public key, *ID* from message details.
- Server generates the anonymous PseudoID, which is a point mapping of the user's unique *ID* on the elliptic curve that initializes the IBES.
- Server generates the individual secret key for the public key *ID*.
- Server decrypts voter's SK_T.
- Server encrypts voter's new secret key and anonymous PseudoID using symmetric encryption algorithm DES and SK_T.
- Server sends secret keys encrypted using SK_T to voter.
- Voter decrypts using SK_T and gets his/her secret key and the PseudoID.

It is assumed that the user uses his/her PseudoID to access the voting system rather than his/her *ID*. Thus, user anonymity is maintained.

Here after, in the rest of the paper, the *ID* refers to the PseudoID, which is a mapped point of the unique *ID*.

5.2 Voter Registration Process

- Voter enters the *ID* and his/her credentials.
- This information is encrypted using Administrator's public key and sent to the Administrator/Registrar.
- Administrator after receiving this information, decrypts using his private component, checks whether the information provided by the voter is valid against the entry in the NRDB.
- If the verifying process succeeds, i.e. the user entry is found in the NRDB.
  - The Administrator generates the passphrase or a Voter Secret Code (VSC).
  - This is encrypted with the public component i.e. the *ID* of the voter and sent to voter.
- If verification fails, a failure message is sent to voter.

The voter after receiving the encrypted message will use his private key corresponding to his public component, *ID* and decrypts the message and gets the VSC.

5.3 Authentication and Ballot Distribution

- Voter sends his *ID* and the VSC by encrypting with public key of the validator.
- Validator decrypts using his private key and checks the eligibility of the voter to cast a vote.
- Validator checks whether the voter has voted before or not (this is to avoid the double voting).
- If voter has already casted vote, then his request to cast a vote is rejected.
- If it is for first time and the voter is eligible to cast the vote, the validator encrypts ballot consisting of the candidate list corresponding to the constituency of the voter using *ID* of the voter.
- There is an option for multiballot selection in this system if a candidate wants to vote for more than two elections.
- The ballot will be identified by the unique ballot identity number.
- The ballot identity number is unique within the constituency and among the constituencies.
- The ballot is sent based on the constituency ID selected by the user.
- The counter is maintained or the ballot ID's helps in checking the number of voters voted in a specific constituency (which is identified by unique constituency).

5.4 Vote Casting

- Voter receives the encrypted ballot and decrypts using his/her private key.
- Voter casts his/her vote by entering *ID*, VSC and selecting candidate of his/her choice. This information is encrypted using collector's public key and sends to the collector.
- The time of the vote cast is stored in the database and an entry is made against the voter as vote casted.

5.5 Collecting, Storing and Counting Votes

- Collector receives the encrypted vote data from the voter, decrypts using his private key and retrieves the *ID*, VSC and vote.
- The encrypted vote is decrypted and the vote count is updated for the corresponding candidate.
- Generates the symmetric key.
- Encrypts the vote with this symmetric key and creates an evote (encrypted vote).
- Generates the new_Anonymous _ID for the voter by creating the hash of ID and the VSC.
- Creates a new_ID by hashing the new_Anonymous _ID and system parameter.
- This system parameter is secret for the entity operating the voting system.
- The symmetric key is encrypted using this new_ID.
- The voter is given with the evote and the encrypted symmetric key.
- The vote information comprising of encrypted vote, new_ID, encrypted symmetric key, and new_Anonymous_ID is stored in the database i.e., vote database.
- The counter does the final tally of the votes and the same is communicated to the Administrator to publish the result.
- The need to store the encrypted vote in the database is to facilitate the verification process.
- The count database represents the result of the voting process as it stores the count of votes each candidate obtained.
- Administrator can publish this result after the election time is over to end the voting process.

6. Security Analysis

In this section we will make some security analysis for our voting system.

- Can an illegal voter cast a valid ballot?

Only a legal voter can register and obtain Voter Secret Code. During the voter registration process, the voter credentials are checked in NRDB. The eligible voters will have an entry in the NRDB. If the voter is not eligible, VSC is not given to the voter. Without VSC voter can't cast a vote.

- Is voter's vote kept private?

The voter's vote is kept private. Though the voting system has the ability to decrypt the encrypted vote, it is able to establish the vote with the particular anonymous Identity and not with any particular voter.

- Is voter anonymity maintained?

As the voter uses PseudoID rather than his/her *ID* to access the voting system, the anonymity of the voter is maintained. The PseudoID is generated by the trusted component and sent to the voter by encrypting using voter's public key.

- Can an illegal voter use another one's credentials to cast a valid ballot?

The voter is provided with a voter secret code. Since voter secret code is secret for the voter and is generated by the voting system for the voter, it is very difficult to obtain the VSC of another person as VSC is generated after verifying the voter credentials.

- Can a legal voter cast more than once?

Legal voter cannot cast vote more than once. During authentication phase the validator checks whether the voter has already casted a vote or not. If the legal voter tries to cast more than once, the voter will be rejected at the authentication phase.

- Can each voter verify whether his/her vote has been counted correctly or not by the voting system?

Yes. A voter can approach the administrator after the results are announced to verify that his/her vote is casted in the way he/she intended. The voter receipt has encrypted symmetric key and the encrypted vote. The voter enters the *ID* and the VSC. The administrator creates the new_Anonymous_ID and the new_ID. The private key d, corresponding to the new_ID is obtained by communicating with PKG. The vote is decrypted using d. The vote information stored in vote database is retrieved, decrypted using d and verified. In this way voter can verify the vote and cannot prove his vote using receipt.

- Is the fairness requirement achieved?

Yes. The results of the elections are published once the timer set to keep track the election time expires. Administrator cannot publish the results until this timer expires the. Hence, the fairness property is achieved in our system.

- Does the voting system avoid vote selling/trading?

Yes. Each voter is given a voter receipt. Voter receipt includes an encrypted vote data which can only be decrypted with the assistance of the entity operating the voting system as the system parameter is secret to the entity which operates the voting system.

- Can voter use his/her receipt to prove his vote?

No. After the voting the voter is given a receipt comprising of encrypted vote (encrypted using the system generated symmetric key) and the encrypted symmetric key (encrypted using new_ID). He cannot decrypt the encrypted vote without the assistance of the entity operating the voting system to any third party. So, voter cannot prove to anyone that he has voted in some way.

7. Implementation

We have used the Eclipse Integrated Development Environment as it provides a faster way of developing the modules and adding library easily. The deployment of the servlet and applets on a web server is easy using Eclipse.

7.1 Cryptographic Libraries

The Internet voting system is implemented using java language version jdk1.6.0_02. The Java Cryptography Architecture (JCA) and Java Cryptography Extension (JCE) are used to provide the cryptographic functions to the entities of the voting system. DES algorithm is implemented using the JCE functions. The Legion of bouncy castle is used as the provider for the Big Integer Operation and Hash functions. Eclipse 3.2 Integrated Development Environment, Apache Tomcat Server tools is used. The servlets are deployed on the apache tomcat server.

7.2 IBE Server as a Downloadable Applet

IBE server is implemented and deployed as an applet providing the functionalities such as, mapping function from *ID* to a point on the curve, function to get the private key given a public Key *ID*, encryption and decryption functions. DES, a Symmetric key algorithm is implemented and deployed as the downloadable applet. Hash functions, random password generation functions are used.

7.3 Entities of the Voting System

All the entities of the Internet voting system are implemented on UNIX system as JAVA servlets and are deployed on Apache Tomcat web server. We have developed the main entities of the Internet voting system by using servlets and java crypto libraries. The voter is provided with the registration page and he/she accesses the voting system through the user friendly web pages. Mysql database is used for maintaining the data of the voters and the voting system. JDBC (Java Database Connectivity) and standard SQL (Structured Query Language) queries are used to access the database.

8. Performance Measurement

The performance is measured on a 3.20 GHz Intel Xeon quad CPU system.

8.1 Server Availability

The availability of the server is tested using the web stress testing tool. The burst of requests are sent to the server running the web application and the number of requests that the server can handle is measured. The encryption and decryption operations are not taken into account at client and server side. Figure 3 shows the results.

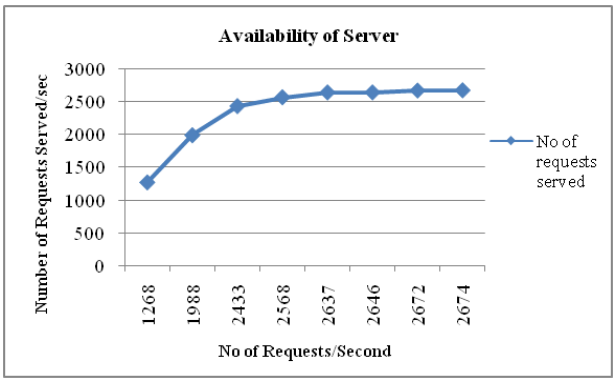

Figure 3. Availability of the server

It can be observed from Figure 3 that the failure rate is very low while serving the requests.

8.2 Computation cost of IBE Server operations

The computation costs for IBE server's setup, private key generation, encryption and decryption

operations when it is initialized with a 192-bit length elliptic curve and 14 digit public *ID* are as depicted in Table 1.

Table 1. Computation cost of server operations.

Server Operations	192-bit curve (ms)
Setup	546
Private Key generation for 14 Digit ID	109
Encryption	507
Decryption	274

The server setup time is not taken into account for measuring the time for Encryption, Decryption and private key generation operations of the server.

8.3 Time Taken for Database Operations

Table 2 shows the time taken for carrying out various database operations.

Table 2: Time taken for various database activities

Operations	Time taken(ms)
Database Connection	359
Database Insertion	2
Database Update	2
Database Read	1

8.4 Theoretical Calculation of Cost/Request

The voter has to carry out the following steps to cast his/her vote successfully.

- Obtaining PseudoID and Secret key
- Voter Registration
- Authentication and ballot distribution
- Casting vote
- Collecting, storing and counting will be carried out by the collector/counter.

Let C_{ps} be the cost to obtain PseudoID and secret key, C_R be the cost for registration process, C_A be the cost for validation and ballot distribution, and C_V for casting vote and C_C storing and counting vote.

Let *msgsize* = average size of each message. We assume the communication cost is *x ms/kilobyte.*

Cost of sending one message will be *msgsize* × *x ms.*

PKG server is initialized with 192-bit curve.

The computation costs are as depicted in Table 1.

8.4.1 Calculation of C_{ps}. The operations carried out for obtaining PseudoID and secrete key are as follows.

1. Voter sends encrypted temporary key and *ID* to server. Server decrypts and gets the key and *ID*.
2. Server encrypts PseudoID, Voter Secret key and sends to voter. Voter decrypts the message and gets the PseudoID and secret key.

There are 2 messages transferred, 2 encryption, 2 decryption computations are performed using PKG Server.

C_{ps} = [computation cost] + [communication cost].

C_{ps}= *(2× msgsize × x + 2 ×507 + 2 ×274) ms.*

8.4.2 Calculation of C_R. The operations that are performed in voter registration process are,

1. Voter sends the encrypted *ID* and credentials to the administrator. Administrator decrypts and gets the details and checks the entries in NRDB.
2. The generated VSC is encrypted and sent to the voter after the successful check of the voter details. Voter decrypts and retrieves the VSC.

There are 2 messages transferred between voter and administrator, 2 encryption and decryption computations are performed using PKG. Thus,

C_R = *(2× msgsize × x + 2 ×507 + 2 ×274) ms.*

8.4.3 Calculation of C_A. The operations performed for authentication and ballot distribution.

1. Voter sends the encrypted *ID* and the VSC to the validator. Validator decrypts and checks for the authenticity of the voter and for double voting.
2. Validator sends the encrypted ballot consisting of the candidate list to the voter. This ballot is encrypted using the *ID* of the voter.

There are 2 messages transferred and 2 encryption and 1 decryption computations are performed.

C_A= *(2×msgsize × x+ 2 ×507 + 1 ×274) ms.*

8.4.4 Calculation of C_V. The operations that account for the cost calculation are as follows,

1. Voter decrypts the encrypted ballot.
2. Casts his/her vote and sends the encrypted vote to collector.

There is 1 message transfer and 1 encryption computation. Thus,

C_V = *(1×msgsize × x+ 1 ×507) ms.*

8.4.5 Calculation of C_C. The operations performed by collector are as follows.

1. The collector receives encrypted vote information from voter and decrypts it.
2. Encrypts vote with symmetric key.
3. Sends the encrypted evote and the encrypted symmetric key by encrypting with voter *ID*.

There is 1 message transfer, 1 encryption, 1 decryption and 1 symmetric encryption computations performed. 26 ms is taken for symmetric encryption.

$C_C = (1 \times msgsize \times x + 1 \times 507 + 1 \times 274 + 1 \times 26)$ *ms.*

Thus, Time taken by server per voting request = $C_{ps} + C_R + C_A + C_V + C_C$.

8.5 Time taken by the server

At an average each message size is 7 kilobytes. We assume the communication cost x, to be 1 ms/kilobyte (this may vary depending on network characteristics). So, the time taken to cast a vote successfully is 5771 ms. Figure 4 gives the time taken by the server to complete voting process.

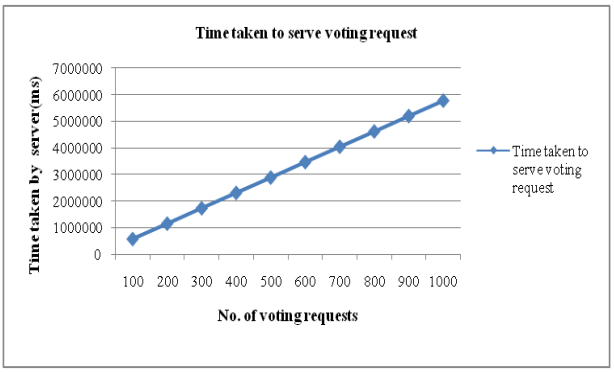

Figure 4. Time taken by server to serve voting request

Approximately, one voting process can be completed per second. It should be noted that the communication cost and computation cost depends on the cost of the environment.

9. Concluding Remarks

We have implemented the Internet voting system prototype which meets the security requirements mentioned in section 3. The participation rate of voting is supposed to increase as voters have an option of using the web based system to cast their vote.

This web based voting system is a new attempt that removes the disadvantages and improves over [7][8].

The implemented voting system is a web based system. Applets are used to implement the security part of the voting system. Applet is a downloadable program that can be executed in a voter's browser that supports Java. The voter's do not need to download any code ahead of time and install it into their client computers.

The developed system uses Identity Based Encryption System. There is no certificate retrieval, nor certificate chain determination and certificate verification as in PKI, which makes the system efficient. The systems remain efficient with IBES.

Security analysis is done to prove that, the system meets the security requirements. The theoretical calculation of the cost per request to vote is carried out. Approximately, one voting process can be completed per second. The computation cost of the PKG server operations and database operations is calculated. Voter anonymity is achieved as PseudoID rather the *ID* is provided to the voting system.

References

[1] Cetinkaya, Orhan Cetinkaya, Deniz, "Towards Secure E-Elections in Turkey: Requirements and Principles", *Proc. IEEE Conference on Availability, Reliability and Security, ARES 2007*, 10-13 April 2007, pp .903 – 907.

[2] Orhan Cetinkaya, "Analysis of Security Requirements for Cryptographic Voting Protocols", *3rd IEEE International Conference on Availability, Reliability and Security*, 4-7 March 2008, pp. 1451-1456.

[3] Dimitris A. Gritzalis, "Principles and requirements for a secure e-voting system", *Computers & Security,* Elsevier *advanced technology*, 1 October 2002, Vol 21(6), pp. 539-556.

[4] Anane R, Freeland R, Theodoropoulos G, "e-Voting Requirements and Implementation", *E-Commerce Technology and the 4th IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services*, 23-26 July 2007, pp.382-392.

[5] Jinn-Ke Jan, Chih-Chang Tai, "A secure electronic voting protocol with IC cards", *Journal of Systems and Software, IEEE publication 1995*, November 1997, pp. 93 – 101.

[6] Kiayias, A.; Korman, M.; Walluck, D., "An Internet Voting System Supporting User Privacy", *IEEE 22nd Annual conference on Computer Security Applications,* December 2006, pp.165 – 174.

[7] Kwangjo Kim, Jinho Kim, Byoungcheon Lee, Gookwhan Ahn, "Experimental Design of Worldwide Internet Voting using PKI", *SSGRR2001*, L'Aquila Italy, 6-10 August, 2001.

[8] Ibrahim S, Kamat M, Salleh M, Aziz S.R.A, "Secure E-voting with blind signature", *Proceedings of the 4th IEEE National Conference on Telecommunication Technology*, 14-15 Jan 2003 , pp. 193-197.

[9] Dan Boneh, Matthew Franklin, "Identity-Based Encryption from the Weil Pairing", *SIAM J. of Computing*, 2003, Vol 32(3), pp. 586-615.

[10] Sharath Palavalli, Srinivas U S. Alwyn R Pais, "Identity Based DRM System with Total Anonymity and Device Flexibility using IBES", *SHPCS08*, Cyprus, June 3-6, 2008.

[11] John R.Vacca, *Public Key Infrastructure Building Trusted Applications and Web Services*, Auerbach publications, Newyork, 2004.