ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

HOMOMORPHIC ENCRYPTION AND RE-ENCRYPTION APPLIED TO VOTING DATA SECURITY

Md. Sohel Ansari [*], Dr. Mukta Bhatele, Prof. Raghvendra Singh Tomar

[*] *Department of Computer Science & Engineering, Jai Narain College of Technology*
Bhopal (M.P.), [INDIA]

ABSTRACT

Homomorphic Encryption is a good basis to enhance the security measures of untrusted systems/applications that stores and manipulates sensitive data. This strong protection of data results from the capability, allowed through HES, to perform arithmetic operations over encrypted bits. The homomorphic property of various cryptosystems can be used to create secure voting systems.

The Idea of the project is create the Web interface for the voters to conduct voting online. However security and integrity of vote count on the data server is a major concern. To secure vote count on data Server to be hacked, the proposed system is to use the homomorphic encryption techniques. Also use of Re-encryption further enhanced Voting Data Security.

KEYWORDS: *Homomorphic Encryption, Re-encryption, Private Key, Serialization, Vote count, FHE.*

INTRODUCTION

Homomorphic encryption is a form of encryption which allows specific types of computations to be carried out on cipher text and generate an encrypted result which, when decrypted, matches the result of operations performed on the plaintext. This is a desirable feature in modern communication system architectures.

Homomorphic encryption would allow the chaining together of different services without exposing the data to each of those services, for example a chain of different services from different companies could 1) calculate the tax 2) the currency exchange rate 3) shipping, on a transaction without exposing the unencrypted data to each of those services. A cryptosystem which supports both addition and multiplication (thereby preserving the ring structure of the plaintexts) is known as fully homomorphic encryption (FHE) and is far more powerful. Using such a scheme, any circuit can be homomorphically evaluated, effectively allowing the construction of programs which may be run on encryptions of their inputs to produce an encryption of their output. Since such a program never decrypts its input, it can be run by an untrusted party without revealing its inputs and internal state. The existence of an efficient and fully homomorphic cryptosystem would have greatpractical implications in the outsourcing of private computations,

Homomorphic Encryption is a good basis to enhance the security measures of untrusted systems/applications that stores and manipulates sensitive data. This strong protection of data results from the capability, allowed through HES, to perform arithmetic operations over encrypted bits.

The homomorphic property of various cryptosystems can be used to create secure voting systems, the Idea of the project is create the Web interface for the voters to conduct voting online. However security and integrity of vote count on the data server is a major concern. To secure vote count on data Server to be hacked, the proposed system is to use the homomorphic encryption techniques. A windows Interface will be provided to the election commissioner that will be used to configure the Election. On configuring Election, the Election commissioner will generated the private Key and it will be serialized on the Computer of the Election commissioner. For all the parties participating in the election, the vote count for each of them will be initialized to Zero and will be stored in the database of the server in the encrypted form encrypted using the generated Private Key by the Election commissioner. On the Election date, the registered voters can login on to the Web Site and can vote for their desired party. When voting is done, one will be added to the encrypted vote count and incremented vote count will be stored on the data server in the encrypted Form.

For enhanced Security, during the voting period, the Election commissioner can use Re-Encryption that is to generate the new Private Key and re-encrypt the previously encrypted data using the newly generated Private Key.

On declaration date, the election commissioner will retrieve the Private Key by DE serializing it. Then the encrypted vote count will be retrieved from Server and using the Private Key will be decrypted and the vote count will be placed on the Server, so that the viewers will be able to view the result. It also provides facility to display the result of Election in graphical format.

LITERATURE REVIEW

Here we will discuss about the findings by study and research that is critical and have an important value in the contribution of the whole project. It also gives some basic knowledge or theoretical base and is used as a foundation to successfully achieve the main objectives. Most of the literatures are from the related articles, journals, books and previous works of the same fields. These literatures are then compiled and use as a guidance to the work of this project.

Here we will provide a brief discussion of voting systems used in current elections. Here we try to develop Online Voting and securing data using Homomorphic and Re-Encryption technique. But it is important to understand the advantages and disadvantages of both paper-based and electronic systems.

Paper-based voting systems

Auditability is the primary argument for paper systems. If ballots are stored safely and securely, there can be as many independent audits as needed. The fact that the audits can be independent is especially important. Each recount involves examining the original ballots, as marked and verified by the voter, rather than relying on a machine's recording of the ballots.

The second major advantage of paper-based ballots, voter verifiability, has become more prominent since the 2000 U.S. presidential election. Many have looked to paper systems to guarantee voters that their ballots were cast as intended because all paper based systems involve permanently marking a piece of paper. After a voter makes her choice, she can visually inspect the paper to ensure the correct choice is indicated. As long as the voter selects a candidate, the vote indicated cannot be changed without invalidating the race or ballot.

The primary types of paper systems are hand-counted, punch-card, and optical scan ballots.

The primary types of paper systems are hand-counted, punch-card, and optical scan ballots. They differ in the method of marking choices and tabulating the results.

The paper systems vary in ease of use and ease of tabulation.

With respect to tabulation, hand-counting is infeasible for elections on the scale of US national elections [24]. It is too slow, expensive, and cumbersome given the complexity of the ballots. However, hand-counting remains a backup method of auditing all paper-based systems. A hand-count of a small statistical sample can trigger a full recount if the distribution of votes differs significantly from that of the electronic or mechanical count.

Electronic Voting Systems:

There are two main groups of supporters for electronic voting systems: voters interested in the convenience and usability of the systems, and election officials interested in a simpler, more flexible, and less costly system. No studies that conclusively demonstrate that electronic voting is more usable exist, mainly because there are so many different systems.

Usability advantages of DREs:

The length of current ballots creates problems for paper-based systems. Elections are rarely a one-race affair and there are typically many more than two candidates for each race. Elections are also used as an opportunity to present referenda on public issues, which are typically written using legal terminology and are difficult to understand. The result ballots that are often double sided and printed in small font sizes. Even with the Federal Election Commission's mandated minimum 6.3 mm character size [9], many elderly and impaired voters are unable to read ballot text.

The length of current ballots creates problems for paper-based systems. Elections are rarely a one-race affair and there are typically many more than two candidates for each race. Elections are also used as an opportunity to present referenda on public issues, which are typically written using legal terminology and are difficult to understand. The result ballots that are often double sided and printed in small font sizes. Even with the Federal Election Commission's mandated minimum 6.3 mm character size [9], many elderly and impaired voters are unable to read ballot text.

An under-vote occurs when a voter does not select a candidate for a race. While it is allowable for a voter to choose not to vote in a race, if the voter casts an under-vote because she did not see the race, it is an error. With large and complicated ballots, these errors are more common.

With DRE machines, races can be presented individually. The voter can be forced to either choose a candidate or acknowledge that she is not voting in the race. This could reduce under-votes because the voter must explicitly choose not to vote in a particular race.

Using a paper-based ballot, a voter can mark multiple selections for a race where only one selection is allowed. This is known as an over-vote. Whether this is due to stray marks or confusion, the result is that the voter's choice is invalidated because election officials are unable to determine the voter's intent. A computer can disallow selecting more than the allowed number of candidates and thereby eliminate over votes.

Another advantage of DREs is the voter's ability to change her ballot without the intervention of election officials. If a voter marks her ballot, then wishes to change her choice, most paper-based systems would require that she turn in her old ballot.

This policy results in a lack of privacy for the voter, who may have only marked one choice incorrectly and is now forced to reveal the rest of her choices. To avoid this, electronic systems allow voters to change their votes without any intervention from election officials. Whether doing so is simple and straightforward depends largely on the user interface.

Disadvantages of DREs:

The major criticism of the DRE voting systems is that they give voters no confidence that the machines are doing the proper thing. After a voter submits her ballot, she has no way of knowing that the machine is recording and counting the vote as entered. To believe this occurred, the voter must trust that the vendors did not intend to miss record votes, that the software developers performed their job competently, that the software was properly certified, and that the machine is running the certified software. This also assumes that the certification standards are high enough to ensure proper security. The problem with trusting the vendors' intent is that the companies making these machines may not be unbiased parties. The companies that produce voting machines, as well as the executives that run those companies, have a history of supporting and donating to particular political campaigns[29]. Furthermore, some of the officials responsible for selecting and regulating electronic voting equipment are elected. There is clearly a conflict of interest in these cases.

In other situations where partisan individuals are responsible for critical electoral processes, efforts are made to disclose their actions as much as possible and to allow members of any political group to participate. One example is the presence of party observers at poll closings. Poll workers, themselves of varying political beliefs, are watched by representatives of any candidates that choose to provide them. Imposing a similar process on the production of voting machines is not feasible.

Vendors claim that suspicions of bias are unfounded because the software must go through a verification process. However, detecting intentionally faulty software is very difficult. For example, a Rice University professor asked computer science students to introduce bugs into a simple voting system and asked other students to examine the code for bugs. Despite a small code base, only 2,000 lines, some bugs went undetected. Compare this to commercial voting systems with over 50,000 lines of code .Even with professionally trained auditors, malicious bugs could go undetected. Beyond the issue of vendor intent is the problem of vendor competency. It is extremely difficult to achieve correctness in software, as evidenced by the bugs discovered in commercial software on a daily basis. While some bugs are to be expected, some of those discovered in current election systems provide very little confidence those writing the software. One of the more publicized such bugs was the hard-coding of keys into the software [30]. This meant that every election district using that software had the same key, and that the key could not be changed without changing the underlying software. These keys were used to encrypt all of the ballots and to set up the memory cards used to authenticate voters. Knowledge of these keys could allow an adversary to cast extra votes, among other things.

One way of reassuring the public of the impartiality and correctness of the voting system is to test the system using predefined standards. Currently, election systems are certified by individual states, based on results from both federal and state tests. These tests generally include auditing the code for errors. The current process is considered inadequate by many, especially because “commercial off the shelf” software is allowed to be included without being audited for errors. Commercial off the shelf software, such as operating systems purchased by vendors from other companies, is used as-is in the voting machines. The current process also treats certification as a one-time process and does not provide an opportunity for citizen involvement or significant public disclosure. The result is that voter confidence is not particularly high. An improved certification process would help improve trust, but examining the code and running tests can never completely ensure correctness, especially if the programmer is malicious.

Another way of improving security and gaining public trust is to require that voting machine software be open source. This solves the problem of transparency by allowing the public to participate in the development process as coders or observers

However, open source based voting machines are not likely to be profitable. A more limited approach would be to make the source code publicly available for evaluation only. In the end, open source cannot completely eliminate errors or malicious code, although it may improve public trust.

Other voting systems have similar problems with achieving trust, but manage to avoid the criticism heaped on DREs because it is possible to recover from fraud by recounting ballots. DRE voting machines have no meaningful recount ability. Optical scan machines use software that is susceptible to the same fraud and correctness errors as DREs, but the ballots are not affected by such errors. Optical-scan ballots can be manually recounted if necessary. In contrast, the only copies of the ballot on a DRE machine are the ones the machine chooses to store. Even if large errors such as obvious candidate bias are detected, no recovery is possible.

I. Need and significance of proposed research work

The main problem with current DRE systems is that they require a large amount of trust from the election officials, who are either elected officials themselves or else appointed by elected officials. However, there has been a significant amount of research on providing cryptographic schemes that reduce this burden of trust.

Homomorphic Encryption:

Homomorphic encryption is naturally suited to election schemes. It allows the votes to be tabulated before decryption, improving privacy. For example, in additive Homomorphic encryption, the sum of two cipher texts is a third cipher text that encrypts the sum of the two original plaintexts.

Voting applications may use additive homomorphism to allow tallying to be done before decryption. With other forms of encryption, all the ballots are dissociated from their identifying pieces of information and then decrypted and tallied. If Homomorphic encryption is used, the tallying can be done while the votes are still encrypted, and the final total can then be decrypted. This effectively hides the contents of the original ballots while providing an publicly computable tally.

Homomorphic Encryption is a good basis to enhance the security measures of untrusted systems/applications that stores and manipulates sensitive data. This strong protection of data results from the capability, allowed through HES, to perform arithmetic operations over encrypted bits.

The homomorphic property of various cryptosystems can be used to create secure voting systems,

Homomorphic encryption is naturally suited to election schemes. It allows the votes to be tabulated before decryption, improving privacy. For example, in additive homomorphic encryption, the product of two cipher texts is athird cipher text that encrypts the sum of the two original plaintexts.

More generally, let ⊥ be an operation, m1,m2 be two messages, and let E[m] represents the encryption of the message m under an encryption scheme. The scheme is homomorphic for the operation ⊥ if you can easily find a cipher text c = E(m1⊥m2) from E(m1) and E(m2). That is, the operation ⊥ can be performed on the underlying messages without revealing them. For election systems, a scheme where ⊥ is a addition is most useful.

METHODOLOGY/ Planning of work

The purpose of the methodology is to give an experienced investigator enough information to replicate the study.

The homomorphic property of various cryptosystems can be used to create secure voting systems,

The Idea of the project is create the Web interface for the voters to conduct voting online. However security and integrity of vote count on the data server is a major concern. To secure vote count on data Server to be hacked, the proposed system is to use the homomorphic encryption techniques.

A windows Interface will be provided to the election commissioner that will be used to configure the Election. On configuring Election, the Election commissioner will generated the private Key and it will be serialized on the Computer of theElection commissioner. For all the parties participating in the election, the vote count for each of them will be initialized to Zero and will be stored in the database of the server in the encrypted form encrypted using the generated Private Key by the Election commissioner .

On the Election date, the registered voters can login on to the Web Site and can vote for their desired party. When voting is done, one will be added to the encrypted vote count and incremented vote count will be stored on the data server in the encrypted Form.

For enhanced Security, during the voting period, the Election commissioner can use Re-Encryption that is to generate the new Private Key and re-encrypt the previously encrypted data using the newly generated Private Key.

On declaration date, the election commissioner will retrieve the Private Key by DE serializing it. Then the encrypted vote count will be retrieved from Server and using the Private Key will be decrypted and the vote count will be placed on the Server, so that the viewers will be able to view the result. It also provides facility to display the result of Election in graphical format.

1. A web front-end that allows Indian citizens to signup, login and update his/her profile as well as register for voting. To register, the user required Aadhar card Number is required that will be verified from an Agency that keeps track of Aadhar card of all Indian citizens.

Also users in the role of an administrator (Election commissioner) will configure election and view party, candidates and voters information.

2. A database which stores the information of all the parties, candidates and voters as well as the election information and vote count in Encrypted Form. After declaration of Result, the result will also be kept in database. A Database module will handle all above said information.

A Window Interface for the Election commissioner for generating Private Key, and generate related Public Key and serialized and DE serialized facility. It is also useful for initializing vote counts and for declaration of Result. Also provide Re-Encryption facility.

Evaluation of Experiment

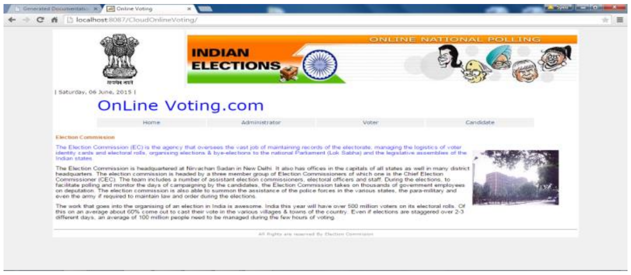
The Idea of the project is create the Web interface for the voters to conduct voting online. However security and integrity of vote count on the data server is a major concern. To secure vote count on data Server to be hacked, the proposed system is to use the homomorphic encryption techniques.

A windows Interface will be provided to the election commissioner that will be used to configure the Election. On configuring Election, the Election commissioner will generated the private Key and it will be serialized on the Computer of the Election commissioner. For all the parties participating in the election, the vote count for each of them will be initialized to Zero and will be stored in the database of the server in the encrypted form encrypted using the generated Private Key by the Election commissioner .

On the Election date, the registered voters can login on to the Web Site and can vote for their desired party. When voting is done, one will be added to the encrypted vote count and incremented vote count will be stored on the data server in the encrypted Form.

For enhanced Security, during the voting period, the Election commissioner can use Re-Encryption that is to generate the new Private Key and re-encrypt the previously encrypted data using the newly generated Private Key.

On declaration date, the election commissioner will retrieve the Private Key by DE serializing it. Then the encrypted vote count will be retrieved from Server and using the Private Key will be decrypted and the vote count will be placed on the Server, so that the viewers will be able to view the result. It also provides facility to display the result of Election in graphical format.



Generated Documentatio
Online Voting
localhost:8087/CloudOnlineVoting/

INDIAN ELECTIONS
ONLINE NATIONAL POLLING

| Saturday, 06 June, 2015 |

OnLine Voting.com

Home | Administrator | Voter | Candidate

Election Commission

The Election Commission (EC) is the agency that oversees the vast job of maintaining records of the electorate, managing the logistics of voter identity cards and electoral rolls, organising elections & bye-elections to the national Parliament (Lok Sabha) and the legislative assemblies of the Indian states.

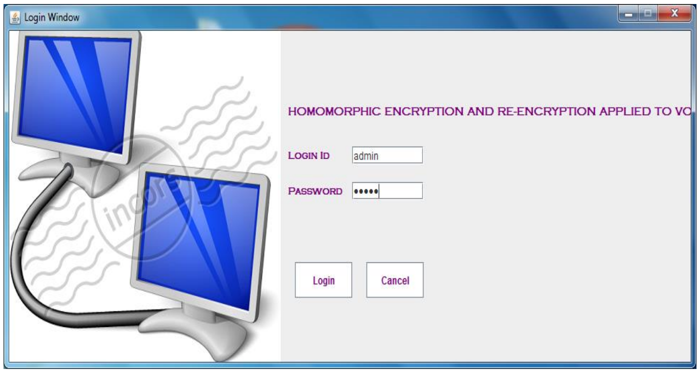
The Election Commission is headquartered at Nirvachan Sadan in New Delhi. It also has offices in the capitals of all states as well in many district headquarters. The election commission is headed by a three member group of Election Commissioners of which one is the Chief Election Commissioner (CEC). The team includes a number of assistant election commissioners, electoral officers and staff. During the elections, to facilitate polling and monitor the days of campaigning by the candidates, the Election Commission takes on thousands of government employees on deputation. The election commission is also able to summon the assistance of the police forces in the various states, the para-military and even the army if required to maintain law and order during the elections.

The work that goes into the organising of an election in India is awesome. India this year will have over 500 million voters on its electoral rolls. Of this on an average about 60% come out to cast their vote in the various villages & towns of the country. Even if elections are staggered over 2-3 different days, an average of 100 million people need to be managed during the few hours of voting.

All Rights are reserved By Election Commission

Figure 1.1 Web Interface for Election commissioner, candidates and voters

Client interface to Administrator, Here Election commissioner will be the Administrator. Using this interface, admin will specify the number of bits used to generate the private and public key which is then serialized. Using these keys, vote count for the parties will be encrypted. And vote count will be decrypted when election result needs to be declared.



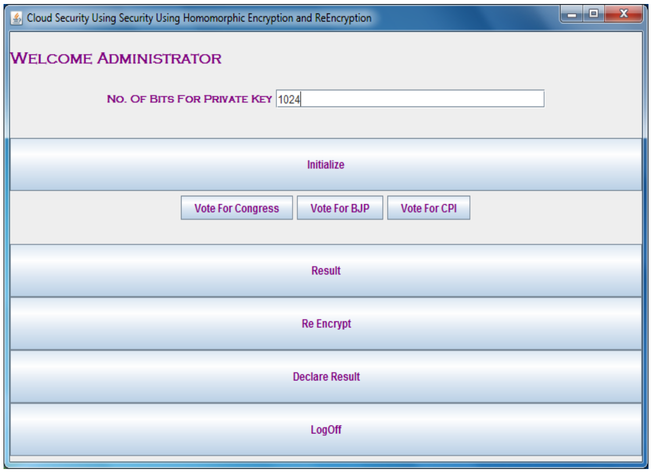
Login Window

HOMOMORPHIC ENCRYPTION AND RE-ENCRYPTION APPLIED TO VO

LOGIN ID: admin

PASSWORD: •••••

Figure 1.2 Login

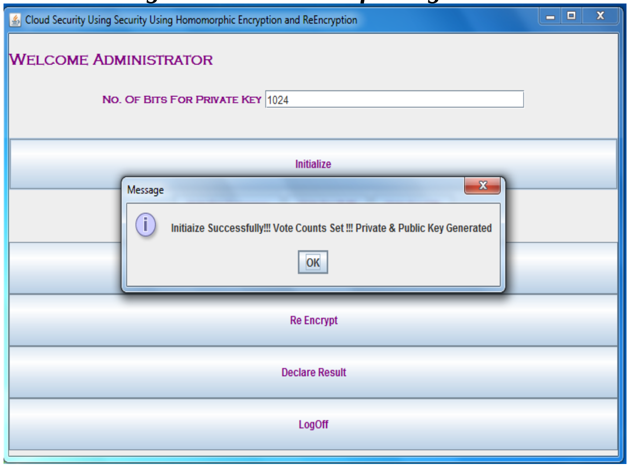


Cloud Security Using Security Using Homomorphic Encryption and ReEncryption

WELCOME ADMINISTRATOR

NO. OF BITS FOR PRIVATE KEY: 1024

Figure 1.3 on successful Login



Cloud Security Using Security Using Homomorphic Encryption and ReEncryption

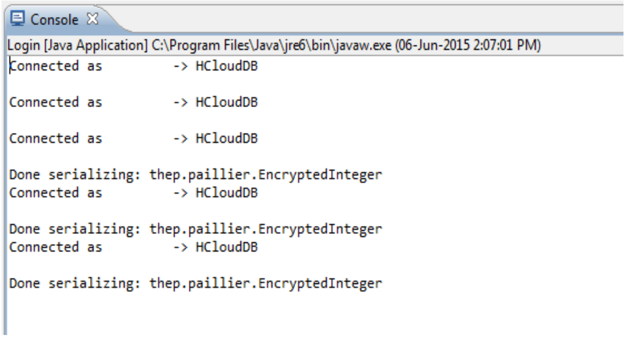
WELCOME ADMINISTRATOR

NO. OF BITS FOR PRIVATE KEY: 1024

Message

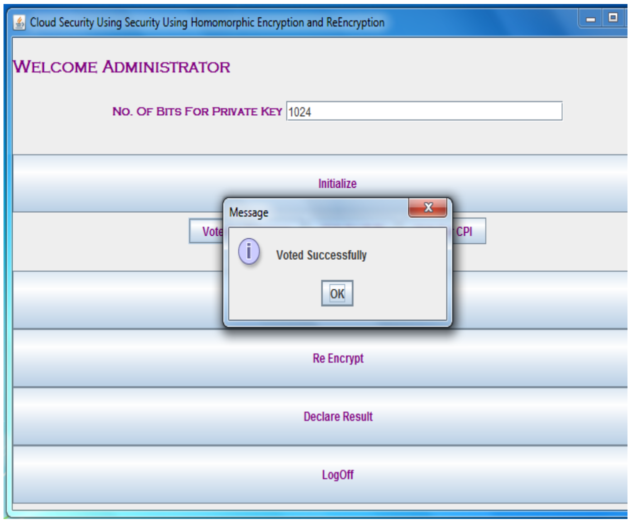
Initiaize Successfully!!! Vote Counts Set !!! Private & Public Key Generated

Figure 1.4 on Successful Initialization



```
Login [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (06-Jun-2015 2:07:01 PM)
Connected as         -> HCloudDB

Connected as         -> HCloudDB

Connected as         -> HCloudDB

Done serializing: thep.paillier.EncryptedInteger
Connected as         -> HCloudDB

Done serializing: thep.paillier.EncryptedInteger
Connected as         -> HCloudDB

Done serializing: thep.paillier.EncryptedInteger
```

Figure 1.5 Status at console Window


Cloud Security Using Security Using Homomorphic Encryption and ReEncryption
Initialize
Vote
CPI
Message
OK
Re Encrypt
Declare Result
LogOff

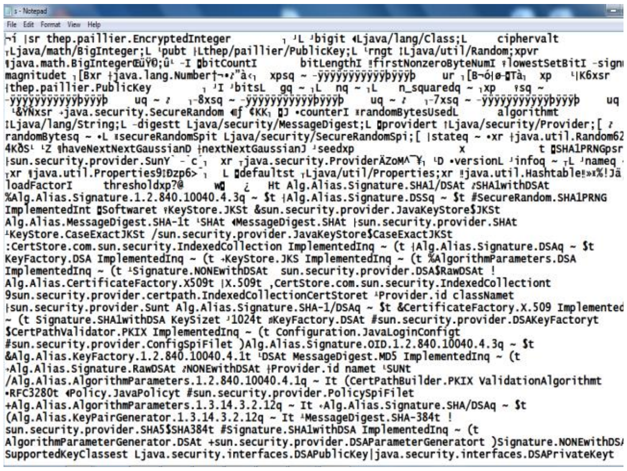
WELCOME ADMINISTRATOR

NO. OF BITS FOR PRIVATE KEY 1024

Voted Successfully

Figure 1.6 Administrator testing for voting

The Encrypted Count: Serialized With java. In actual application, it is serialized as a blob in oracle database. If viewed, will be displayed as follows:



```
¬í |sr thep.paillier.EncryptedInteger        ┐ ┘L ┘bigit ◄Ljava/lang/Class;L     ciphervalt
┬Ljava/math/BigInteger;L └pubt ├Lthep/paillier/PublicKey;L └rngt ↕Ljava/util/Random;xpvr
¶java.math.BigIntegerŒüŸ©;û└ -I ▯bitCountI       bitLengthI ‼firstNonzeroByteNumI ♀lowestSetBitI -sign
magnitudet ┐[Bxr ┼java.lang.Number†¬•♪"à‹┐  xpsq ~ -ÿÿÿÿÿÿÿÿÿÿÿþÿÿÿþ    ur ┐[B¬ó|ø▯Tà┐  xp   └|K6xsr
┤thep.paillier.PublicKey          ┐ ┘I ┘bitsL  gq ~ ┐L  nq ~ ┐L   n_squaredq ~ ┐xp   ♀sq ~
-ÿÿÿÿÿÿÿÿÿÿÿþÿÿÿþ    uq ~ ♪   ┐-8xsq ~ -ÿÿÿÿÿÿÿÿÿÿÿþÿÿÿþ    uq ~ ♪   ┐-7xsq ~ -ÿÿÿÿÿÿÿÿÿÿÿþÿÿÿþ    uq
 └&ŸNxsr ┼java.security.SecureRandom ◄f ¢KK┐ ▯J •counterI ♯randomBytesUsedL     algorithmt
↕Ljava/lang/String;L -digestt Ljava/security/MessageDigest;L ▯providert ↑Ljava/security/Provider;[ ♪
randomBytesq ~ •L ♯secureRandomSpit Ljava/security/SecureRandomSpi;[ |stateq ~ •xr ┼java.util.Random62
4KðS└ └Z ¶haveNextNextGaussianD ┼nextNextGaussianJ ┘seedxp             x             t ▯SHA1PRNGpsr
┤sun.security.provider.SunY` - `c`┐  xr ┬java.security.ProviderÄZoM^¯¥┐ └D •versionL ┘infoq ~ ┬L ┘nameq
┬xr ¶java.util.Properties9↕Ðzp6>`┐  L ▯defaultst ┬Ljava/util/Properties;xr ‼java.util.Hashtable‼»♯%!Jä
loadFactorI    thresholdxp?@     w▯   ¿   Ht Alg.Alias.Signature.SHA1/DSAt ♪SHA1withDSAt
%Alg.Alias.Signature.1.2.840.10040.4.3q ~ $t ┤Alg.Alias.Signature.DSSq ~ $t #SecureRandom.SHA1PRNG
ImplementedInt ▯Softwaret ♀KeyStore.JKSt &sun.security.provider.JavaKeyStore$JKSt
Alg.Alias.MessageDigest.SHA-1t └SHAt ◄MessageDigest.SHAt ├sun.security.provider.SHAt
┴KeyStore.CaseExactJKSt /sun.security.provider.JavaKeyStore$CaseExactJKSt
:CertStore.com.sun.security.IndexedCollection ImplementedInq ~ (t ┤Alg.Alias.Signature.DSAq ~ $t
KeyFactory.DSA ImplementedInq ~ (t ⇢KeyStore.JKS ImplementedInq ~ (t %AlgorithmParameters.DSA
ImplementedInq ~ (t ┴Signature.NONEwithDSAt  sun.security.provider.DSA$RawDSAt !
Alg.Alias.CertificateFactory.X509t |X.509t ,CertStore.com.sun.security.IndexedCollectiont
9sun.security.provider.certpath.IndexedCollectionCertStoret ┴Provider.id classNamet
┤sun.security.provider.Sunt Alg.Alias.Signature.SHA-1/DSAq ~ $t &CertificateFactory.X.509 Implemented
~ (t Signature.SHA1withDSA KeySizet ┘1024t ♯KeyFactory.DSAt #sun.security.provider.DSAKeyFactoryt
$CertPathValidator.PKIX ImplementedInq ~ (t Configuration.JavaLoginConfigt
#sun.security.provider.ConfigSpiFilet )Alg.Alias.Signature.OID.1.2.840.10040.4.3q ~ $t
&Alg.Alias.KeyFactory.1.2.840.10040.4.1t └DSAt MessageDigest.MD5 ImplementedInq ~ (t
⇢Alg.Alias.Signature.RawDSAt ♪NONEwithDSAt ┼Provider.id namet └SUNt
/Alg.Alias.AlgorithmParameters.1.2.840.10040.4.1q ~ It (CertPathBuilder.PKIX ValidationAlgorithmt
•RFC3280t ◄Policy.JavaPolicyt #sun.security.provider.PolicySpiFilet
+Alg.Alias.AlgorithmParameters.1.3.14.3.2.12q ~ It ⇠Alg.Alias.Signature.SHA/DSAq ~ $t
(Alg.Alias.KeyPairGenerator.1.3.14.3.2.12q ~ It ┴MessageDigest.SHA-384t !
sun.security.provider.SHA5$SHA384t #Signature.SHA1withDSA ImplementedInq ~ (t
AlgorithmParameterGenerator.DSAt +sun.security.provider.DSAParameterGeneratort )Signature.NONEwithDS/
SupportedKeyClassest Ljava.security.interfaces.DSAPublicKey|java.security.interfaces.DSAPrivateKeyt
```

Figure 1.6 Encrypted Count

CONCLUSION

The system is developed as a Web Site. The developed application will be deployed on a space purchased on a web server. The homomorphic property of various cryptosystems can be used to create secure voting systems.

The Idea of the project is create the Web interface for the voters to conduct voting online. However security and integrity of vote count on the data server is a major concern. To secure vote count on data Server to be hacked, the proposed system is to use the homomorphic encryption techniques.

A windows Interface will be provided to the election commissioner that will be used to configure the Election. On configuring Election, the Election commissioner will generated the private Key and it will be serialized on the Computer of the Election commissioner. For all the parties participating in the election, the vote count for each of them will be initialized to Zero and will be stored in the database of the server in the encrypted form encrypted using the generated Private Key by the Election commissioner. On the Election date, the registered voters can login on to the Web Site and can vote for their desired party. When voting is done, one will be added to the encrypted vote count and incremented vote count will be stored on the data server in the encrypted Form.

For enhanced Security, during the voting period, the Election commissioner can use Re-Encryption that is to generate the new Private Key and re-encrypt the previously encrypted data using the newly generated Private Key.

On declaration date, the election commissioner will retrieve the Private Key by DE serializingit. Then the encrypted vote count will be retrieved from Server and using the Private Key will be decrypted and the vote count will be placed on the Server, so that the viewers will be able to view the result. It also provides facility to display the result of Election in graphical format.

Advantages:

1. Voting Simplified.
2. Use of Homomorphic Encryption ensures secured Voting.
3. It prevents unregistered users or unregistered users from using voting.
4. Ensures biased free Voting.

REFERENCES

[1] MS. PARIN V. PATEL#1, MR. HITESH D. PATEL*2, PROF. PINAL J. PATEL#3, A Secure Cloud using Homomorphic Encryption Scheme, International Journal of Computer Science Research & Technology (IJCSRT) Vol. 1 Issue 1, June –2013

[2] Maha TEBAA, Saïd EL HAJJI, Abdellatif EL GHAZI ,Homomorphic Encryption Applied to the Cloud Computing Security, Proceedings of the World Congress on Engineering 2012 Vol WCE 2012, July 4 - 6, 2012, London, U.K

[3] Payal V. Parmar, Shraddha B. Padhar, Shafika N. Patel, Survey of Various Homomorphic Encryption algorithms and Schemes,International Journal of Computer Applications (0975 – 8887), Volume 91 – No.8, April 2014

[4] Md. Sohel Ansari, Prof. Raghvendra Singh Tomar, Prof. B. L. Rai, Dr. Mukta Bhatele, Homomorphic Encryption and Re-Encryption Applied to Voting Data Security, International Journal of Modern Engineering & Management Research, Volume 3 Issue 1 | March 2015