CHAPTER 39
Implementation of Advanced E-Voting Application Using Android Platform

Dr. V. Jayaraj
Nehru Institute of Engineering and Technology, Coimbatore, India
Mrs. K. Sivakami
Nehru Institute of Engineering and Technology, Coimbatore, India
Mrs. S. Priya
Nehru Institute of Engineering and Technology, Coimbatore, India
Dr. S. Arunkumar
Nehru Institute of Engineering and Technology, Coimbatore, India

ABSTRACT

The project is to develop an Online E-Voting prototype system utilizing the Paillier Threshold Cryptosystem (PTC) web services and applying MESE processes to it in an attempt to find possible solutions to further improve existing PTC web services. Online voting (e-voting) would be more convenient, relatively secure and utilize fewer resources. To be able to access e-voting system from a personal, business or even a public library computer may be more convenient for many people needing to vote. This could potentially be a solution for the low voter turnout at the polls. However, it is still questionable whether elections can be conducted online or over the internet due to the high level of concern over security. Systems considered to be apart of e-voting are Machine readable (create, read, count) ballot systems, Direct Recording Electronic (DRE) systems, voting using mobile devices and internet voting .As part of this project, an online e-voting prototype system has been constructed using the demonstration windows application tool created for PTC web services. A pre-computation process is applied due to efficiency improvements. The details of this optimization and improvement in the web services process will be explained in the subsequent sections. In addition to the application of the pre-computation to the process, the Chinese Remainder Theorem can be applied during the decryption process. This change might not be as noticeable as the pre-computation, however it will make it more efficient as the calculation gets easier.

Keywords: *Relatively secure and utilize fewer resources, Ballot systems, Direct Recording Electronic (DRE) systems,*

INTRODUCTION

In this project, PTC Web services are used. In this section, I will explain how the PTC web services work. Efficiency improvement that will be applied to the PTC web services required some changes on some of

ISBN: 978-93-5782-920-5, www.aetsjournal.com —Handbook of research in Big data Analytics, Artificial intelligence and Machine learning

the classes used. Applying more improvements will need more changes on the classes where calculations applied. Details will be explained in the following sections of this report.

ONLINE E-VOTING SYSTEM PROJECT DESCRIPTION

PAILLIER THRESHOLD CRYTOSYSTEM WEB SERVICES ARCHITECTURE AND DESIGN

The Paillier cryptosystem is a probabilistic asymmetric algorithm for public key cryptography, first published by Pascal Paillier in 1999. This probabilistic scheme has generated a good amount of interest and further study since it was discovered.

The problem of computing n-th residue classes is believed to be computationally difficult to compute. This is known as the Composite Residuosity (CR). The scheme is an additive homomorphic cryptosystem; this means that, given only the public-key and the encryption of m_1 and m_2, one can compute the encryption of m1 + m2.

One of the properties of Paillier as mentioned above is the homomorphic property which can allow this cryptosystem to do simple addition operations on several encrypted values and obtain the encrypted sum. The encrypted sum can later be decrypted without ever knowing the encrypted values that made up the sum. Due to these useful characteristics of Paillier, the scheme has been suggested for use in threshold cryptosystems, secret sharing schemes and the design of voting protocols especially the evoting systems.

Another property of Paillier cryptosystem is self-blinding. This property is essential as it means a ciphertext can be re-encrypted with a random parameter without changing the underlying cleartext and without changing the ability to decrypt the ciphertext using the original keypair[15]. Probabilistic property of Paillier will help to protect voter’s privacy since none of the votes will be encrypted to the same ciphertext.

Paillier has described three different methods in his research. PTC Web services that will be used in this project are using one of these three methods. Below are the schemes invented by Pascal Paillier and

Scheme 1: Scheme 1 is probabilistic encryption scheme based on composite residuosity. According to theorem mentioned in his paper [21] Scheme 1 is one-way if an only if the Computational Composite Residuosity Assumption holds. It is also semantically secure if and only if the Decisional Composite Residuosity Assumption hold. n is the multiplication of two prime numbers, n = pq. g is randomly selected base. This can be doneby checking whether $\gcd\left(\mathrm{L}(g^{\lambda} \bmod n^2), n\right) = 1$. This is done on the PTC web services used. n and g are public parameters and (p, q) or λ remains private.

Encryption: plaintext m < n randomly select r < n

ciphertext c = $g^m \cdot r^n \bmod n^2$

Decryption:

ciphertext c < n2

plaintext $m = \dfrac{\mathrm{L}(c^{\lambda} \bmod n^2)}{\mathrm{L}(g^{\lambda} \bmod n^2)} \bmod n$

Table 3.1 Paillier’s Scheme 1 [21]

Scheme 2**:** Scheme 2 is a trapdoor permutation based on composite residuosity. As described above n is the product of two prime numbers. From the table below, there are steps explained for decryption. To be able to retrieve m, all these steps will be required.

Scheme 2 is one-way if and only if RSA [n,a] is hard .

Encryption: plaintext m < n2

split m into m1, m2 such that m = m1 + nm2

ciphertext c = $g^{m_1}{m_2}^{n} \bmod n^2$

Decryption:

ciphertext c < n2

Step 1. $m_1 = \dfrac{\mathrm{L}(c^{\lambda} \bmod n^2)}{\mathrm{L}(g^{\lambda} \bmod n^2)} \bmod n$

Step 2. $c' = cg^{-m_1} \bmod n$

Step 3. $m_2 = c'^{\,n^{-1} \bmod \lambda} \bmod n$

plaintext m = m1 + n m2

Table 3.2 Paillier’s Scheme 2 [21]

Scheme 3: Third scheme is the variant with fast decryption. As this is a fast decryption, this scheme can be applied to improve the efficiency. In the following sections this scheme will be re-visited and it will be recommended for efficiency improvements in the current web services.

Encryption: plaintext m < n randomly select r < n

ciphertext = $g^{m+nr} \bmod n^2$

Decryption:	ciphertext c < n2 $\text{plaintext } m = \frac{\mathrm{L}(c^{\alpha} \bmod n^2)}{\mathrm{L}(g^{\alpha} \bmod n^2)} \bmod n$

Table 3.3 Paillier's Scheme 3 [21]

It is assumed that g Є $\mathcal{B}_{\alpha}$ for some 1 ≤ α ≤ λ. In other words α and λ are not the same secret keys.

Below are the steps for the key generation, encryption and decryption used .

Key generation

1. Choose two large prime numbers *p* and *q* randomly.
2. Compute *n* = *pq* and λ = *lcm*(*p* − 1, *q* − 1)
3. Select random integer *g* where $g \in \mathbb{Z}^*_{n^2}$
4. Ensure *n* divides the order of *g* by checking the existence of the following

multiplicative inverse: $\mu = (L(g^{\lambda} \mod n^2))^{-1} \mod n$

where function *L* is defined as $L(u) = \frac{u-1}{n}$

The public (encryption) key is (*n*,*g*).

The private (decryption) key is (λ,μ).

Encryption

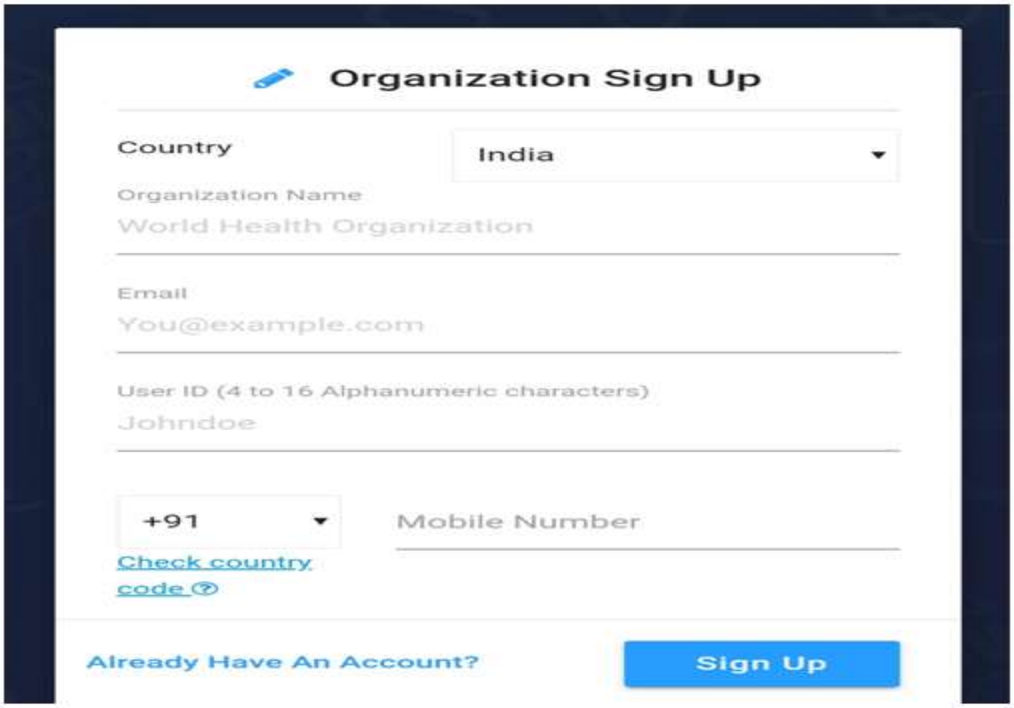
1. Let *m* be a message to be encrypted where $m \in \mathbb{Z}_n$
2. Select random *r* where $r \in \mathbb{Z}^*_{n^2}$
3. Compute ciphertext as: $c = g^m \cdot r^n \mod n^2$

Decryption

1. Ciphertext $c \in \mathbb{Z}^*_{n^2}$
2. Compute message: $m = L(c^{\lambda} \mod n^2) \cdot \mu \mod n$

It is the same as the scheme 1 described above. This computation takes some time due to the large prime numbers used. The secret key is SK = λ(n) = lcm((p-1),(q-1)). BB

SCREENSHOTS


Organization Sign Up

Country: India

Organization Name: World Health Organization

Email: You@example.com

User ID (4 to 16 Alphanumeric characters): Johndoe

+91 | Mobile Number

VOTING PROTOTYPE SYSTEM

The capabilities of the Paillier Threshold Cryptography system has been demonstrated on an Online E-Voting Prototype system created for this project. This is a prototype and should not be used in the real world scenarios. It shows the use of the Paillier Threshold Cryptography Web Service. It also has some additional security features like Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) added to decrease the security concerns.

E-VOTING SYSTEM OVERVIEW

The e-voting system allows for 1 out of L candidate ballots. No options are provided for n out of L ballots or write-in ballots. An “election” may consist of more than one ballot. An election administrator creates the ballots and other election parameters. The administrator requests the Paillier threshold encryption parameters from the PTC Web Service during the initial election set-up. The administrator submits the election parameters to a VotingService web service, and saves the election parameters (including the cryptosystem parameters) to an XML file. Voters then load the election parameters by opening the XML file, make their selection(s), and cast their encrypted vote(s) to the VotingService web service. During the tally phase, the votes are multiplied together, and, due to the homomorphic properties of the Paillier cryptosystem, the product can be decrypted to reveal the sum total of all the votes .

USER LOGIN

User Login is the first form users connected when the voting page is loaded from the internet. It will have a connection to the database to validate the user credentials. User types are either voters or Administrators. It is assumed that users have used another interface or form to register for voting. In the same login page there will be Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) validation with random numbers. Six digit random numbers will be created each time the page is loaded to be able to stop any kind of computer attacks to the voting site.

ELECTION SET-UP

The election administrator uses the Election Builder form to create or modify an election (before the election is posted to the voting web service). To create a new election, the administrator clicks on the “New Election” button. A new election is created and a unique election id is assigned. The administrator must then enter his/her name and a descriptive title for the election. Election page is the most important Administrator page as it has all the functionality setup for the election.

Before ballots can be added to the election, the encryption parameters must be specified and retrieved from the web service. This must occur before the ballots are added or created, since the vote format is dependent on the specified key size. The administrator clicks to the “Encryption Parameters” . This button will be available after the Administrator details are entered. Once this button is clicked, the administrator specifies the key size and whether or not to encrypt the returned key shares. The administrator can then add the key share owner information for each owner that is to receive a secret key share. If the key shares will be encrypted, the administrator will be required to enter the owner’s username which is the same as the users login and certificate name to be able to choose automatically. Once all owners have been added, the administrator selects the cryptosystem threshold value and then clicks “Send Request”, which sends the request to the web service. In the current configuration, a key size of larger than 256 and sometimes 512 bits will result in such a delay that a “timeout” error is caused, so it is not recommended that key sizes greater than 256 be used for the web application. The web service will generate the requested parameters, encrypt the key shares (if specified), and return them . The Encryption

Parameter Request form will transfer the returned parameters to the Election Builder form and close automatically. The election crypto parameters are displayed at the bottom of the Election Builder form.

Lastly, on the same election page ballots can be added for the election. If the ballots are created prior to the election creation page, the list will appear in the window for administrator to choose from the list. They can be added to any election by highlighting from the list and clicking to the”Add Ballots” button. If the ballot is valid, it will be imported into the election and displayed in the Election Summary textbox in the form.

After all the users, ballots and Administrator details loaded from the election form, the Administrator will need to save and post the election to be able to initialize election voting. The election will be saved as an XML file. First save the election by clicking to the

“Save Election” button. It will be saved in the web server “App_Data/XMLFiles Folder”. Details of the folder structures are documented in the Software Design Specification document. Posting the election to the voting web service is a non-reversible operation in the application unless the details are manually deleted from the database. Post Election button will be enabled after saving the election. To post the election, click to “Post Election” button. A web service call will be made that posts the election data to the web service, which then creates the appropriate database entries that are used to manage the election .

CREATING BALLOTS

Existing ballots can now be added to the election or new ballots can be created using the options from the Election form. To create a new ballot, the administrator will need to click to the “New Ballot” link from the elections page. It will open the Ballot Builder form. A new ballot will be created and the random ballot id displayed in the form. Administrator will need to put ballot issue/ problem, and then enter all of the available choices, one at a time by using the “Add Choices” button and the text box. Each choice is entered by typing the appropriate text. A choice can be deleted by selecting the choice in the list, and clicking “Delete Candidate” button. When the ballot is complete, the ballot should be saved by clicking “Save ballot” button. This button will get all the details entered and save the ballot in XML format in the web server “App_Data\XML Files\Ballots” folder. The Ballot Builder Form must be closed and then re-opened in order to create another ballot. Ballot creation page is also accessible from the Administrator menu.

VOTE FORMAT

When a ballot is added to an election, the format of the vote for that ballot is derived from the key size chosen for the election and the number of “candidate” choices on the ballot. These two values determine the maximum number of voters allowed. The total size of the vote is limited to the key size k (in bits). The vote is split into c bit fields where c is the number of candidates. The size of the bit fields vc= k/c. However, vc is limited to 32 bits so that each candidate’s field will fit into a 32-bit integer (for ease of extraction only). Therefore, if k/c> 32, vc=32 and only the first 32*c bits of the vote will be used. To cast a vote, a voter votes the value 2^(ic*vc) where ic is the desired candidates ballot index (0,…,c-1). By using votes of this format, the tally can be computed by multiplying all of the votes together and decrypting the product. Due to the homomorphic property of the Paillier cryptosystem, the multiplication carried out in the ciphertext space corresponds to addition in the cleartext space, and thus the decryption of the product will contain the summed votes for each candidate. Each candidate’s bit field can then be extracted and evaluated to determine the total number of votes for that candidate .

VOTING

CREATING THE VOTE

Once an election has been created, saved, and posted to the election web service, voters can create and cast votes. After the user login page user logs in either as an Administrator or a voter. If the user logs in as an Administrator, he will have a link from the menu for the voting page. If the user has logged in with voter credentials, then he will be connected to the voting page automatically. When connected to the voting page, a list box will have all the elections available for the voters. This list is the list of the elections in the elections folder. After highlighting the election and clicking to the button to load the election, election details will be loaded for voters to vote. The ballots from the election will be loaded, with each issue being loaded into the issue text box, and it’s corresponding choices loaded into the textbox to the right (the choices textbox). The voter can make his/her choice simply by clicking on the desired choice. That issue’s choices will then be displayed in the choices textbox. Again, select the desired choice by clicking on it in the choices textbox. Once a choice has been selected, the ballot issue and the selected choice will appear in the “Current Votes” textbox. To the right of the issue question and the selected choice is the hex value of the vote to be cast. Once all choices have been made, the voter can submit his/her vote by selecting “Submit Vote” button at the bottom of the page. This button will cal the web services and save the vote into the database. Once the vote is submitted, no changes can be made.

At any time after submitting his/her vote, a voter can check the posted values of his/her vote by selecting "Check Submitted Vote" button. This invokes a web service call to the voting web service which retrieves the encrypted vote values posted for that election .

TALLY THE VOTE

Administrator will have access to use the Tall Vote option during the election process to tally the vote. Administrator will need to click the "Tally/Decrypt Vote" button on the menu. The Tally form will open. In a list box elections list will appear for Administrator to choose and tall the vote. If the secret key shares were encrypted, the program will automatically get the certificates according to the issued names of the users to decrypt the owner's Paillier secret key share. That's why it is important for Administrator to collect all the registration details from the user to be able to create the users. He/she will assign the right certificates so that there won't be any issues in the future process like tally / decrypt vote process. The product of the votes for each ballot is then calculated and displayed both encrypted and decrypted, and the candidate's tallies are extracted from the decrypted bit field and displayed.

PRE-COMPUTATION

This change will be done for the key generation where the prime numbers will be calculated prior. Any real-time computations will slow down the process on cryptography application. Any pre-computation will improve the efficiency of the application. This precomputation can be done via background thread setup in the application.

```
<settingname="ServerPath"serializeAs="String">

<value>c:\inetpub\wwwroot\EVoting\PreComputation\</value>

</setting>

<settingname="PrimeNumberCalculationType"serializeAs="String">

<value>DB</value>

</setting>
```

This pre-computation is applied to the SafePrimeNumbers generator function. This function is used for the pre-computation.

CHINESE REMAINDER THEOREM (CRT)

Chinese Remainder Theorem is one of the most useful theorems of number theory as it says it is possible to reconstruct the integers in a certain range from their residues module a set of pair wise relatively prime module. Details of CRT is explained in the previous sections. Paillier has suggested to use CRT for especially key generation and decryption processes [21]. Also CRT has become standard today in many RSA applications as it increases the decryption up to 4 times [16]. Decryptions can be made faster by separately computing the messages mod p and mod q instead of mod n and recombining modular residues later.

With pre-computations:

where p-1 and q-1 have to be placed by α

5.3 Paillier Scheme Pre-computations for Decryption

Scheme 1 used in this project is not the most efficient one especially for decryption as it is also mentioned in Pascal papers study [21]. Scheme 3 improves the performance of decryption and he suggested in the same paper to pre-compute the constant instead of only p and q values applied in this project. Also another constant parameter below can be pre-computed [21].

These constant pre-computations can be done with the same methods used in this project.

RESULTS

PRE-COMPUTATION PERFORMANCE EVALUATIONS

Pre-computations results are put into both the text file and the Pre-Computation tables created in the SQL Server. Both the text file and the database solutions have increased the performance in other words response time more than 80% in average for both 256 and 128 bit key sizes. Unfortunately this test failed with 1024 and 512 bit key sizes due to time out issues.

There is a parameter setup in the settings to use the random prime number generator either real time or text file or database. As a default it will set to the real time. XML solution also needs some improvements and this will be suggested in the future improvements section of the project.

DEFECTS FOUND

These defects are listed in the order in which they were found. It only includes those defects found while creating the automated test suites, not those found and fixed during software development.

DefectID 1: When the election is created, it can not save title and username details in the xml file.

Solution: _election parameter stored in the session was not initialized in the beginning of the function. After initializing it is fixed.

DefectID 2: Back button is required after the ballots are created.

Solution: After ballots are created, back button is required by the Administrator to be able to complete the election creation or ballot creation. Two link buttons are added, one to the Main menu link and the other one is a link to the Elections page.

DefectID 3: Outside the compiler application was not able to respond to the certificate assignment for the users.

Solution: This is fixed by assigning ports each time we run the application. A dedicated port needs to be used by the administrator.

DefectID 4: XML output for the pre-computation does not work properly and need to be fixed. Only real time computation and DB computations work which is enough to show the efficiency improvements in the code.

Solution: This need to be fixed in the future releases.

DefectID 5: User Login page does not hide the password text.

Solution: This is fixed by changing the text box property.

DefectID 6: User Name is the same as the certificate issued name used in the certificate. If these names do not match, certificate can not be used and this will throw an error. To minimize the issues, user name from the login page will be passed to the voting page automatically. This enhancement needs to be applied as this is an additional requirement.

Solution: This is done by using Sessions in ASP .Net. username session is created and the username is passed to the next form which is voting form.

CONCLUSION AND FUTURE SCOPE

Online E-voting system is a prototype developed by using PTC Web services. As the need for voting system has started to increase and some organizations or countries has started to look for the solutions, this can be the starting point to improve and deploy in the real world scenarios.

In this project I have tried to explain the importance of Paillier cryptosystem, , its unique properties and its application areas especially in e-voting.

We need to keep in mind htat voting is not the only process during the whole voting processes. There might be some other security concerns that need to be considered when such an application is built for practical reasons.

Lastly, Paillier Cryptosystem efficiency can be improved as suggested in many papers . Random numbers pre-computation is one of the ways implemented in this project. It has increased the calculation more than one of the ways. In the next section, I will be listing all improvements that can be done to this web service and application.

In this project E-Voting Online prototype application has been implemented. PTC Web Services are used for the encryption and decryption process. The method implemented and used on the PTC Web services is the first scheme invented by Paillier ad explained above. In the following years in numerous projects other similar method called Second Paillier Cryptosystem is used and this calculation simplifies the decryption. This can be implemented in PTC Web services to improve the efficiency.

Additionally, there are few suggestions made about the efficiency improvement above. Any of these or all of these can be applied to make the web services more efficient. Most of the suggestions involve pre-computation of the constants in the schemes invented. The pre-computation applied in this project can be applied to more generic constants and have a dll application running continuously on the back ground thread from the server instead of a button from the web server.

Lastly, tests failed on 512 and 1024 bit key size encryption. Design can be changed to make it work with these key sizes.

REFERENCES

[[1] Dr.Aree Ali Mohammed and Ramyar Adbolrahman Timour,Efficient E-voting Android Based System, IJARCSSE,vol.3,Issue 11,2013

[2] A.S. Belenky and R.C. Larson, “To Queue or not to Queue?,” OR/MS 27, October 2013, pp. 30-34.

[3] “An Electronic Polling Service to Support Public Awareness Using Web Technologies”, Christos Bouras, Nikolaos Katris, Vassilis Triantafillou.

[4] “E-voting on Android System” paper (International Journal of Emerging Technology and Advanced Engineering) prepared by : Kirti Autade, Pallavi Ghadge, Sarika Kale ,Co-authors-Prof. N. J. Kulkarni, Prof. S. S. Mujgond, February 2012.

[5] “Electronic Voting,” Encyclopedia of Computers and Computer History, prepared by Lorrie Faith Cranor and edited by Raul Rojas, published by Fitzroy Dearborn, 2001.

[6] “Voting – What is, What Could be,” Caltech/MIT Voting Technology Project (VTP) Report, July 2001.

[7] Java Cryptography an e-book by Jonathan B. Knudsen, First edition May 1998, ISBN:1-56592-402-9