Abstract

The Philippines has been adopting the Automated Election System after years of adhering and relying on the Manual Voting System. It has been proven that the process became more efficient and generates results in a lesser time frame. However, complications are still evident in present time. For instance, the possibility of electoral fraud continues to persist through the existence of a secret server in the last presidential election. This study aims to propose a technical solution that would eliminate the possibility of secret servers by implementing a public key infrastructure as security measures for the transmission of votes on the server-level.

INTRODUCTION

Background of the Problem

After years of having a manual voting system, the Philippines have finally adopted an Automated Election System (AES) in 2010. This was mandated by Republic Act No. 9369 which is the Amended Elections Automated Law. The law stated that there would be paper-based election system defined as “a type of automated election system that uses paper ballots, records, and counts votes, tabulates, consolidates, canvases, and transmits electronically the results of the vote counts.” (Angkaya, 2011).

The Automated Election Systems in the Philippines follows a sequence of process. Focusing on the transmission module of the system, the process starts when the polling precincts closes on the election day. The PCOS machines transmit the vote counts or election returns to the corresponding servers and canvassing centers. From the PCOS machines, the ERs are transmitted to the central server, to a transparency server, and to the municipal board of canvassers (MBOC) in the consolidation and canvassing system. From the MBOC, the results are brought to the provincial board of canvassers (PBOC) or the regional board of canvassers (RBOC), where the results are collected and then transmitted to the national board of canvassers (NBOC), where the results for national positions are canvassed. The MBOC and PBOC also separately send ERs to the central server. Moreover, the canvassing system processes the ERs that were transmitted. Public telecommunication networks are assigned to be the main channel when transmitting the ERs. There are back-up plans made if the network fails to accomplish its task. For example, transmissions can be made via satellite to avoid delays when problems occur.

During the transmission of ERs, electoral fraud is inevitable. There are a lot of issues concerning the effectivity and efficiency of the machine and servers during the elections. For instance, the camp of Senator Ferdinand Marcos Jr., who ran for the position of vice president, expressed his concerns about the alleged “Secret Servers”. Smartmatic’s Marlon Garcia, the head of the technical support team, admitted that aside from the three servers that was authorized by the comelec, there was also a “meet-me room” where several servers were housed. It has also been further disclosed that the secret servers were intentionally not mentioned by the Comelec and Smartmatic to the public making it more suspicious. The purpose of the said “secret server” or the “queue server” is to first receive all the transmitted votes before distributing it to the three official servers declared by the Comelec. However, the normal process that should be done for the ER transmission as mandated by the law is to simply transmit the votes directly to the said servers. These servers are the Municipal Board of Canvassing Server, the Central Server, and the Transparency Server. Moreover, another problem being pointed out was that the “secret server” never undergone the initial source code review which is essential in every machine that would be used during the elections. Additionally, there were no watchers assigned in the said server putting the integrity of the May 2016 election under a cloud of doubt.

Amidst the controversy of the existence of “Secret Servers”, there has also been a dispute on the altering of the source code resulting to a regeneration of hash codes. From a leaked screenshot, it has been evident that Smartmatic and Comelec uses the MD5 hash code algorithm and at the same time shows that the hash codes mismatched after the alteration was made. However, IT experts claim that the use of MD5 is questionable when it comes to its reliability in securing the data. MD5 is open source and it has been used to verify the file’s integrity.

Statement of the Problem

How can the Philippine automated election system eliminate the possibility of secret servers to secure the transmission of election returns on the server-level?

Objectives

General

* To know the vital issues currently present in the transmission of the votes
* To propose a system that would prevent electoral fraud in the transmission of votes in the automated election system

Specific

* To provide a technical solution that allows the verification of the servers’ authenticity using a public key infrastructure as a security mechanism

Significance of the Study

The findings of this research will benefit the following key players:

To the Filipino Citizens

This study will benefit the Filipino citizens for ensuring the security of the casted votes. Also, it would prevent malicious individuals from manipulating the votes. Although the nature of automation easily provoke fear to ignorance of using technology, this study will educate some of those users that still lack computer literacy foundation. Indeed, education can bridge the existing gaps and even remove the unnecessary fear from automation. This will leave the citizens better equipped for the future of the Philippine automated election system.

To the COMELEC

This research would significantly contribute to the goal of the COMELEC to conduct a fair and transparent election. Considering the impact of the elections in the overall condition and future of the Philippines, it is important to make sure that the voters’ choice reflect the outcome of the election. To do that, the system should be able to prevent and mitigate electoral fraud while ensuring that the voters have casted their votes in a way that is convenient and voter-friendly. This study will aim to determine the most appropriate methods to achieve the kind of system that does not manipulate the vote of the people in any way through data gathering and research. In this manner, the people will be knowledgeable about how the system works and be informed and wise voters themselves.

To the Future Researchers

As the Philippines adjust to this kind of voting system, more and more developers would also contribute to the AES aspect of software development. In that case, the system that would be created can serve as a guide and inspiration for other developers who would want to pursue the prospect of automated election system too.

Scope and Limitation

The scope of the study would only include the issues and possible solutions for the security of the transmission of election returns on the server-level of the automated election system in the Philippines. Further study on the other parts of the automated election system will no longer be covered.

REVIEW OF RELATED LITERATURE

Related Literature

**Automated Election System**

On election day, as the polls close, the BEI immediately administers the transmission of the votes or election returns via the PCOS machines equipped with modems to the servers and canvassing centers The Electronic Results Transmission Service is responsible for the transmission of the votes. The primary channel used is through the public telecommunications networks and if that fails transmission will then be run through the satellite. Furthermore, a software called the Real-time Election Information System, reads the data and canvasses the votes. After the transmission from the PCOS machine, the ERs are transmitted to the central server, to a transparency server, and to the municipal board of canvassers (MBOC). Those three are the official servers declared by the administrators or officials of the elections. Moreover, the MBOC transmits it to the provincial board of canvassers (PBOC) where they consolidate and later transmit the results to the national board of canvassers (NBOC). Additionally, both the MBOC and PBOC sends ERs to the central servers.

There was a special case in ARMM wherein they establish the regional board of canvassers (RBOC). The results for ARMM governor, vice governor, and assemblymen are tallied before being transmitted to the central server.

During the 2016 elections, congress will have their own server wherein the members of the senate and house of representatives can monitor the canvassing of the votes and to officially proclaim the winner for the national level. (Retrieved on August 27, 2016 / <http://www.rappler.com/newsbreak/iq/91663-philippine-automated-election-sytem-explained>).

**Issues in the system**

During the 2016 elections, Bong Bong Marcos’ camp believes that there exists a “Fourth Server” or also known as the “Queue Server”. It has been revealed that the Comelec and Smartmatic has been keeping it from the public. Instead of letting the ERs be directly transmitted to the three official servers, namely the CCS, Central Server, and the Transparency server, the results were first being processed and consolidated in the “Queue Server”. Another problem with the secret server is that the source code being used was never reviewed despite of it being a requirement in the law. Moreover, there were no poll watchers assigned for these servers making it questionable to both the public and the administrators. According to a statement made by Rodriguez, a representative of Marcos’ camp, the integrity of the 2016 May elections has been questioned because of the unexpected situation made by the Comelec and Smartmatic. The ERs that were presented to the public did not come directly from the transparency server. Alternatively, the results were first transmitted to a “Queue Server” where they were “consolidated and processed” and the “Queue Server” sends the data to the transparency servers which is against the law. (Retrieved on August 10, 2016/ <http://www.manilatimes.net/smartmatic-admits-using-unofficial-servers/275442/>)

Officially, there are three servers namely, the transparency server, the central server, and the canvassing center servers. When a voter fills out a ballot it would be fed into the Vote Counting Machine (VCM). The data entering the machine would not be sent to its corresponding servers until the end of the voting period. The votes are then transmitted to the three servers that are independent with each other. This is to secure the data and to make cheating difficult to attackers. It is impossible to hack all 3 servers that are not connected to each other because the results would always be different. Changing the data in one server is not going to change or update the other two and to rigged the election results all three servers would have to be breached to make those results valid. (Retrieved on August 10, 2016/ <https://kami.com.ph/9430-6-things-filipinos-know-vote-counting-machine-issue.html>).

**System Vulnerabilities**

The MD5 Security Algorithm

In 1992, MD5 (Mekle-Damgard 5) security algorithm was developed to address the problem of the MD4 algorithm. It is known to be open source and is widely use to ensure the data’s integrity. However, after 4 years of its development, weaknesses with the said algorithm were discovered upon the researches done by a computer scientist based in the University of California. Moreover, in 2007, Lenstra and Stevens, both are computer scientists, further showed that the MD5 is not a reliable algorithm to use because they found out that the hash function is vulnerable of collisions. This means that two different files with different functions may produce identical MD5 hash values which shows that MD5 is not enough to secure the data’s integrity. The Carnegie Mellon University Software Engineering Institute also rejected the MD5 security algorithm after discovering that attackers of the system can generate data that illegitimately appears to be authentic. They stated that the MD5 algorithm should be considered cryptographically broken or unsuitable for further use. If the MD5 failed to meet its purpose, there are other hashing algorithms that can be used to verify the integrity of the files. (Retrieved on August 28, 2016/ <http://www.thinkingpinoy.net/2016/05/bbm-vs-leni-comelec-smartmatic-obsolete-MD5-technology.html>).

On the other hand, Microsoft, a well-known software company, took a small step to increase the security of enterprises by following industry standards that weaker/shorter key lengths were no longer viable for production use. Microsoft announced a Security advisory that will block the MD5 hash algorithm. This hashing algorithm is quite long in the tooth and has not been a recommended hash for many years. (Retrieved on August 28, 2016 / <https://pkisolutions.com/goodbye-md5-sooner-than-you-think/>).

Several researchers at the Chaos Communication Congress in Berlin showed the weakness of MD5 in which the same hash value was generated with two different files. Although it has been known that it is possible to generate the same hash value, this has been the first time to be demonstrated in just a matter of three days using 200 Sony Playstation3 consoles. It is known a while back that it would take a few years for it to happen. Most industry companies have already been discouraging the use of MD5 for some time and promoting the use of stronger hashing algorithms such as SHA1. These new exploits will certainly push developers away from MD5 to avoid further complications. (Retrieved on August 28, 2016 / <http://www.maravis.com/is-it-goodbye-md5/>).

Hash Codes

The Source code is a sequence of programming code typed by a computer programmer and is readable to humans. It is then converted into a machine readable form known as a compiled or executable program and is dependent to the source code. This is the reason why there should be a source-code review before being converted into an executable program. Each source code is handles by the MD5 command that generates hash codes which serves as the digital fingerprint. The generated hash code is the assurance that whatever was been tested would be the same for the machines used for the elections. If someone changes even a single line of code, the resulting hash code would be different. (Retrieved on August 28, 2016 / <http://www.thinkingpinoy.net/2016/05/bbm-bongbong-marcos-leni-robredo-comelec-hash-code-cheating-math.html>).

PKI

Public Key Infrastructure has been around for quite a while. However, the method of using a public key and a private key in relation to an encrypted message exchange seems straightforward enough, and yet, it has taken a long time for PKIs to become commonplace. The PKI is known for its complexity in managing certificates and keys which needs to be considered. Implementing and managing a PKI is consequently a task not to be taken lightly, and one which will require both commitment and an appropriate level of expertise. Despite its reliability in securing data, many organizations are still hesitant to implement it because they believe that certificate and keys could somehow take care of themselves because of the growing change about IT securities. As with many technological concepts, the key to wider acceptance lies with ease of implementation and usage. Historically, this has been a downfall for PKI which has often been perceived as over complicated and resource heavy from a management perspective. Wider adoption increases familiarity of course, but even so, many might struggle with the detail of managing an organization wide PKI. Wider scale implementations require very careful consideration. August 28, 2016 / <https://www.reconnaissance.net/secure-document-news/issues/april-2016/>).

Related Study

Experimental Design of Worldwide Internet Voting System using PKI

In this study, the researchers designed an Internet voting system applicable for worldwide voting which was based on Ohkubo et al.’s scheme combined with Public Key Infrastructure. In the system, voter’s privacy was guaranteed by using blind signature and mix-net, and robustness which was provided through the threshold encryption scheme. A way of typical implementation for internet voting system was proposed by employing Java technology. PKI allowed worldwide key distribution and “one certificate/one vote” policy. Therefore, anyone can participate if the certificate was given by Certificate Authority (CA). By the joint work between Korean and Japanese teams of this study, the implementation aimed to select MVPs in 2002 FIFA World Cup Korean-Japan in easy and friendly manner for any internet user to participate. (Retrieved on August 27, 2016 / <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=311B92E00249A90FA1A9557F7E3ABA46?doi=10.1.1.6.1111&rep=rep1&type=pdf>). In this study, it was showed it is possible to incorporate a PKI in a voting system. Not only it was possible, it was also recommended for security purposes of the system.

THEORETICAL BACKGROUND

A Public Key Infrastructure is a combination of software and procedures providing a means for managing keys and certificates and using them efficiently. Key and certificate management is the set of operations requires to create and maintain keys and certificates. One of the major points being addressed in a managed PKI is the creation of keys and certificates. A PKI must offer software support for key pair generation as well as certificate requests. Furthermore, there must be procedures to verify the identity of the user before allowing him to request a certificate. Next major point is private-key protection. These private keys are either used for decryption or digital signature so it must have some reasonable level of protection. In this case, a strong password mechanism must be implemented to have an effective PKI. Now, if the user’s private-key has been compromised, the PKI must provide a means by which a certificate can be revoked. Backup and recovery, key and certificate update, and key history management are included in the major points to be addressed if a well-managed PKI is to be implemented. Retrieved on August 27, 2016 / http://www.cgi.com/files/white-papers/cgi\_whpr\_35\_pki\_e.pdf).

In Asymmetric algorithm or Public key cryptography, two keys are used which is known as the public and private keys. One key is provided to cipher the data and the other key is provided to decipher the data. The key that is used to cipher data is publicly known but the other key that is used to decipher is a secret which means that the receiver holds the secret key and using this secret key the receiver can crack the encrypted texts sent by the others. So, a protected information cannot be accessed as long as there is no key. To ensure the authentication, these keys must be verified and current. This algorithm uses hard math problems for the keys mainly by factoring two large numbers since computers are capable of multiplying large numbers but cannot easily factor the product therefore, it will take time to break the key. Furthermore, its main concerns are the confidentiality, integrity, authentication and non-repudiation of information. The advantage of using this algorithm is that if one of the keys has been compromised then the other one is not affected and the guessing of the key algorithm is monitored. With this, it will be easier to distinguish an intruder. (Retrieved on August 31, 2016/https://cseweb.ucsd.edu/~mihir/cse207/w-asym.pdf) .

The commonly used protocol is the Diffie-Hellman that is under the asymmetric algorithm. In this encryption protocol, two people who have not contacted each other before can communicate by sharing a secret key to use for encryption. This protocol is a one-way function which means that it is easy to encrypt but difficult to decrypt because to match the same key there is a mathematical formula that must be followed. Moreover, Diffie-Hellman is known to be used for avoiding attackers from interrupting the transportation of information between two persons. In implementing Diffie-Hellman, two end-users must mutually agree on positive whole numbers of two variables, such that one variable is a large prime number and the other variable is the generator of that prime number. Both end-users must choose a secret number then the user will compute for the public number. After computing for the public numbers, exchange of public numbers will take place. To finish, the computation of traded public numbers will result to their shared key without worrying about the other users obtaining this information. In conclusion, Diffie-Hellman is an effective protocol because of the way it protects a temporary key for communication session. (Retrieved on September 1, 2016/ http://searchsecurity.techtarget.com/definition/Diffie-Hellman-key-exchange).

Hash codes can also be called hash values, hash sums or simply hashes, but not hashish. Hash codes are produced by having a computer ingest any size of data and generating out a small set of hexadecimal numbers. For example, a hash code of “The Future of the Philippines” is 3c57-0b7c-a2d5-fc89-3cde-71d0-cd16-7412. Hash codes can be useful in the society that we have today. For instance, police officers use it as a forensic tool to capture criminals. It can also be used by photographers and songwriters to protect their works for plagiarizers. Additionally, IT professional use hashes to secure and protect their files and verify the data that they’ve been receiving.

An example of hash code algorithm is the MD5. In MD5, the probability of having an identical hash code is 1 in 340,282,366,920,938,463,463,374,607,431,768,211,456. Mathematicians believe that the algorithm of MD5 is weak because they have theoretically demonstrated that they can produce collisions wherein the same hash codes are produced for two different files or data. As a result, people have been switching to SHA-256 that produce hashes that unique for every 115, 792, 089, 237, 316, 195, 423, 570, 985, 008, 687, 907, 853, 269, 984, 665, 640, 564, 039, 457, 584, 007, 913, 129, 639, 936 instances.

The advantage of using a hash code is that it is an excellent detector for tampering and fraud. Although, the change of hash code is not a strong proof that there indeed is a crime committed. Instead, it can only be a trigger in investigating.

Furthermore, hash codes cannot be decrypted because they are strictly one way operations. The only way to hack a hash code is to try a large possible number of inputs and hope for a match. An example was made by Drexx Laggui, principal consultant of Laggui & Associates Inc that conducts vulnerability assessment, Internet penetration testing and computer forensics, if you want to crack a system password stored as an MD5 hash code, you’ll need to produce an MD5 hash of every possible password you think you know and then compare each of those hash codes you have against the stored password hash code. If you find a match, then you can be certain that your guessed password is the correct password. (Retrieved on September 1, 2016/ http://opinion.inquirer.net/94849/hashcode-5-things-you-need-to-know).

PROPOSED SOLUTION TO THE PROBLEM

The current system only provides a hash value that would secure the integrity of the data. However, it lacks the security mechanism that would address confidentiality of the system which makes it possible for other servers aside from the official servers to exist in the transmission of the election returns.

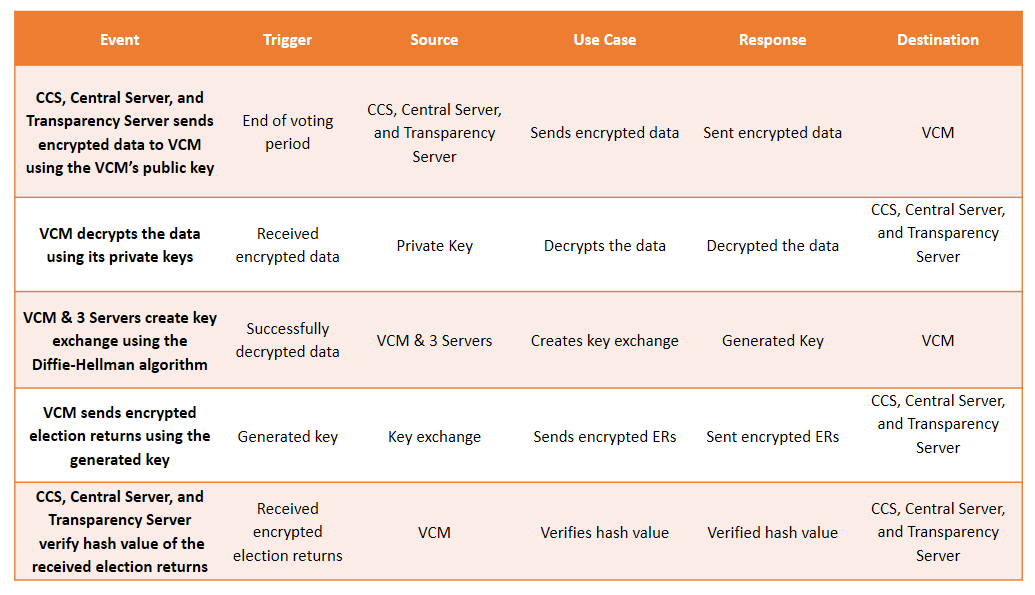
As for the proposed solution to this issue, a hybrid public key infrastructure has to be implemented to not only authenticate the data being transmitted but to also allow the verification of the servers’ authenticity. The process will start once the voting period ends at 5:00pm of the election day. On the current system, the voting counting machine (VCM) will have to immediately send the election returns to the following servers: Municipal Board of Canvassers Server in the Consolidation and Canvassing System (CCS), Central Server, and Transparency Server. In the proposed system, the three said official servers will have to send an encrypted data to the VCM first with the use of the public key of the said VCM. This is to provide the VCM a go-signal to transmit the election returns. The VCM then will have to decrypt the data using its own private key to verify that the data was indeed from an authentic server in the system. Once the identification of the servers is verified, the VCM and the server can now create key exchange using the Diffie-Hellman algorithm that will yield its own generated key. Once a key is generated, this will be used to encrypt the election returns and send to the servers. The servers will have to verify the hash value of the election returns to check the integrity of the data as well as its origin.

RESULTS AND DISCUSSIONS

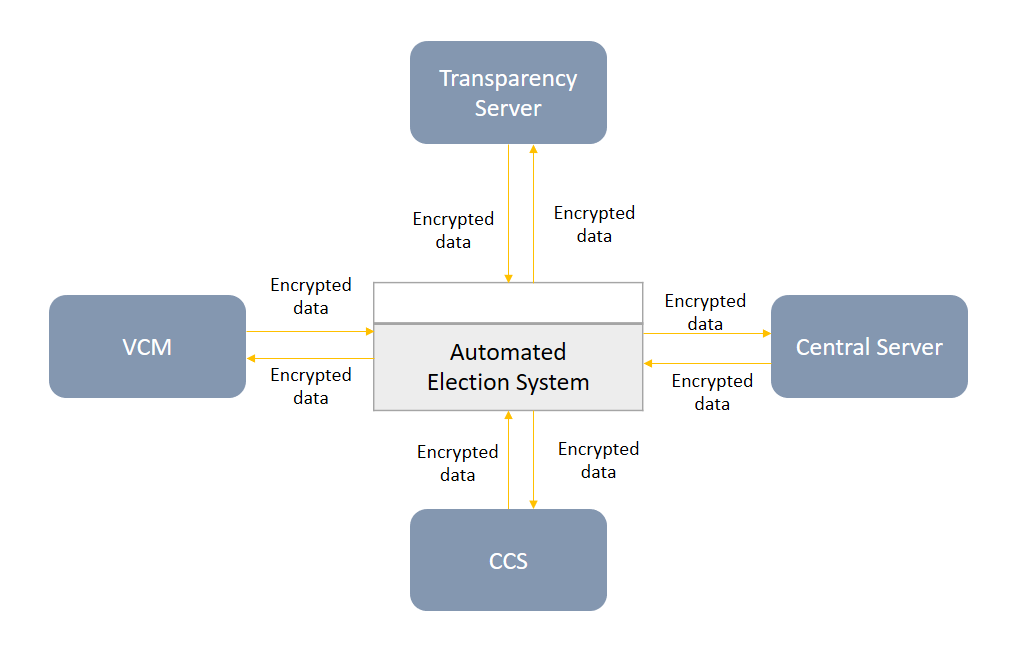
CONCLUSIONS AND RECOMMENDATIONS

APPENDICES

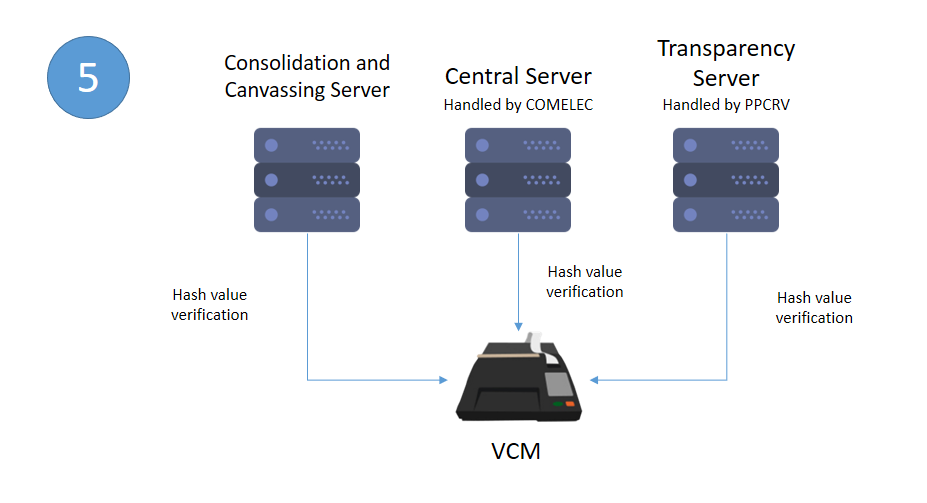
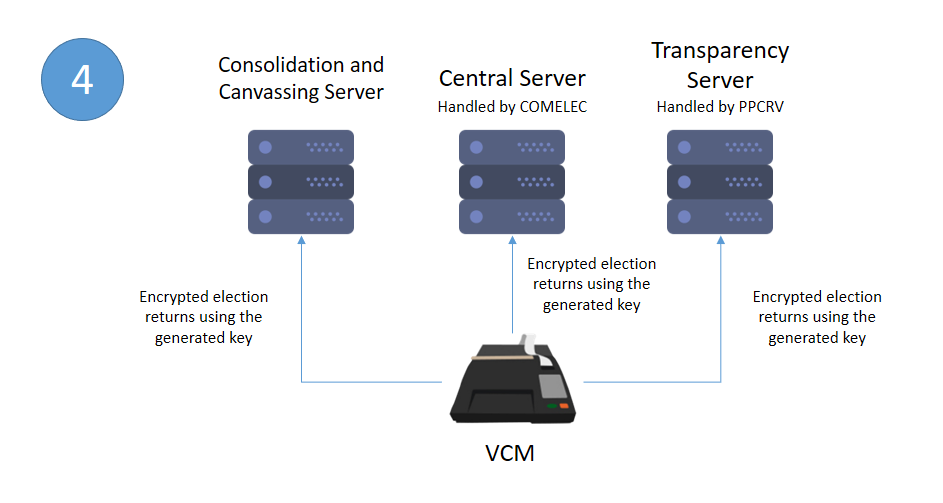
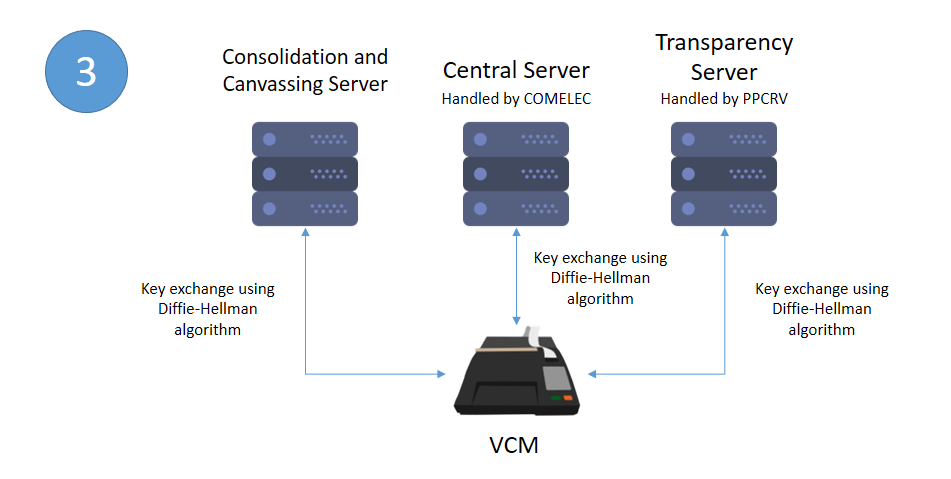
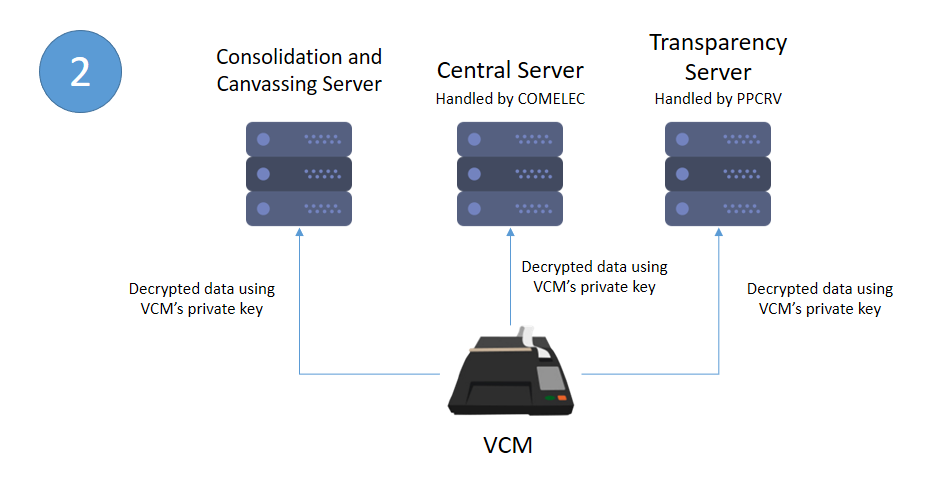
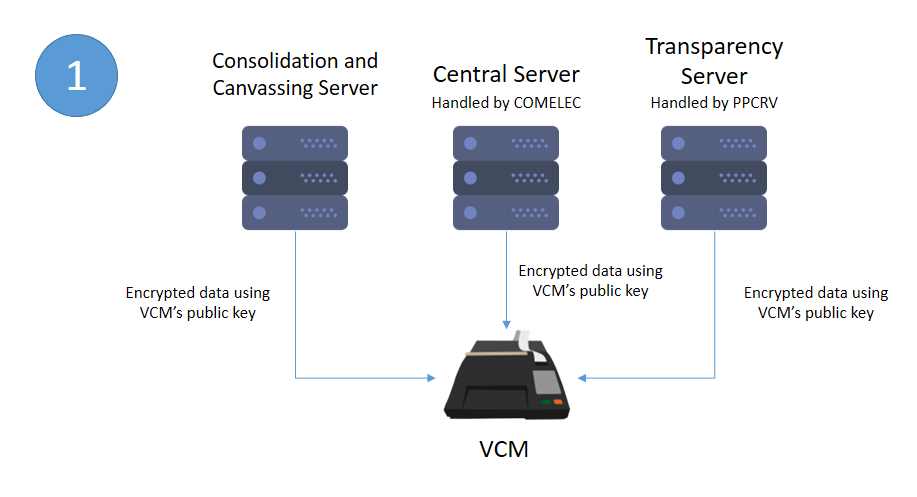
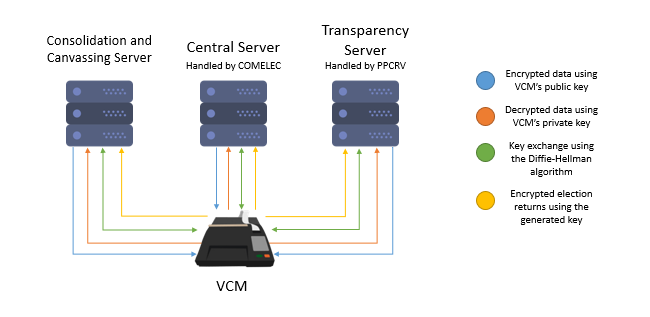
Event Table



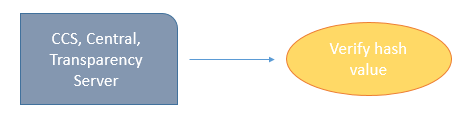
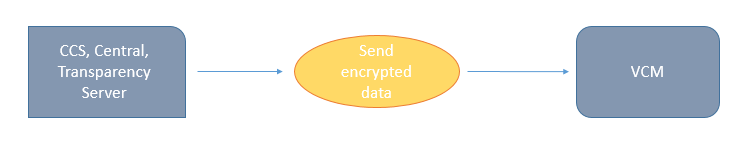
Context Diagram



Data Flow Diagram



Use Case



Use Case with Full Description

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Send encrypted data | |
| Scenario: | CCS, Central Server and Transparency Server sends encrypted data to VCM using the VCM’s public key | |
| Triggering Event: | End of voting period | |
| Brief Description: | The three main servers send a message to the VCM that is encrypted using the VCM’s public key | |
| Actor(s): | VCM, CCS, Central Server, Transparency Server | |
| Related Use Cases: | --- | |
| Stakeholders: | The VCM will receive the data sent by the servers  The three servers will have to send the encrypted data | |
| Precondition: | Voting Period closes | |
| Postcondition: | Sent encrypted data | |
| Basic Flow: | Actor:  1. The Comelec closes the voting period | System Response:  1.1 The CCS. Central Server, and the transparency server sends a message to the VCMs to initiate the vote transmissions. |
| Exceptions: | The machines being used have defects making it impossible to communicate | |

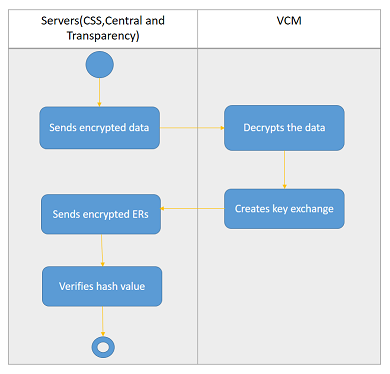
|  |  |  |
| --- | --- | --- |
| Use Case Name: | Decrypts the data | |
| Scenario: | VCM decrypts the data using its private keys | |
| Triggering Event: | Received encrypted data | |
| Brief Description: | In order for the VCM to read the data sent by the servers, it has to decrypt it using its very own private key. | |
| Actor(s): | VCM, CCS, Central Server, Transparency Server | |
| Related Use Cases: | Send encrypted data | |
| Stakeholders: | The VCMs must decrypt the data using the private key assigned to it | |
| Precondition: | Sent encrypted data | |
| Postcondition: | Decrypted the data | |
| Basic Flow: | Actor:  1. Poll watchers ensure that no one will intervene the transmission process. | System Response:  1.1 The System will decrypt the data that was encrypted and interpret it into a language that is understandable. |
| Exceptions: | The machines being used have defects making it impossible to communicate.  Human intervention is present and the system fails to serve its purpose. | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Creates key exchange | |
| Scenario: | VCM & 3 Servers create key exchange using the Diffie-Hellman algorithm | |
| Triggering Event: | Successfully decrypted data | |
| Brief Description: | A security algorithm is used by the VCM and servers in order to generate their own key. | |
| Actor(s): | VCM, CCS, Central Server, Transparency Server | |
| Related Use Cases: | Decrypts the data | |
| Stakeholders: | The VCM and servers must create a key | |
| Precondition: | Decrypted the data | |
| Postcondition: | Generated Key | |
| Basic Flow: | Actor:  1. Poll watchers ensure that no one will intervene the transmission process. | System Response:  1.1 VCM and secret servers will communicate with each other  1.2 They will be able to generate a key exchange. |
| Exceptions: | The machines being used have defects making it impossible to communicate.  Human intervention is present and the system fails to serve its purpose. | |

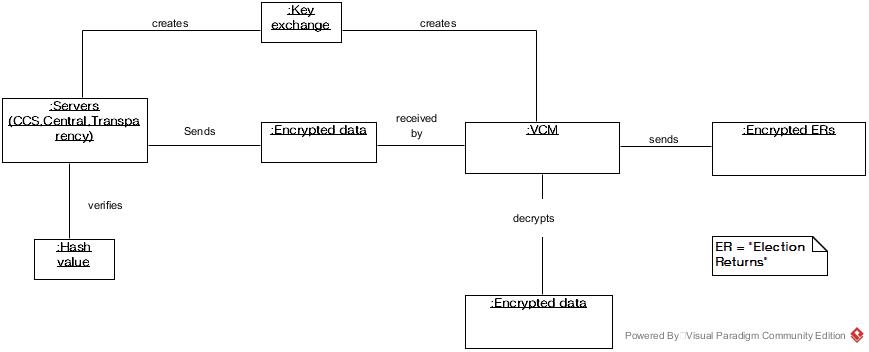
|  |  |  |
| --- | --- | --- |
| Use Case Name: | Sends encrypted ERs | |
| Scenario: | VCM sends encrypted election returns using the generated key | |
| Triggering Event: | Generated Key | |
| Brief Description: | After the key exchange, the VCM will transmit the ERs that were encrypted using the generated key. | |
| Actor(s): | VCM, CCS, Central Server, Transparency Server | |
| Related Use Cases: | Creates key exchange | |
| Stakeholders: | VCM should transmit encrypted ERs to the servers that communicated to it. | |
| Precondition: | Generated Key | |
| Postcondition: | Sent encrypted ERs | |
| Basic Flow: | Actor:  1. Poll watchers ensure that no one will intervene the transmission process. | System Response:  1.1 The VCM will transmit encrypted votes to the servers.  1.2 Servers will receive them |
| Exceptions: | The machines being used have defects making it impossible to communicate.  Human intervention is present and the system fails to serve its purpose. | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Verifies hash value | |
| Scenario: | CCS, Central Server and, Transparency Server verify hash value of the received election returns | |
| Triggering Event: | Completion of Precinct Votes | |
| Brief Description: | Received encrypted election returns | |
| Actor(s): | VCM, CCS, Central Server, Transparency Server | |
| Related Use Cases: | Sends encrypted ERs | |
| Stakeholders: | Servers will determine if the hash values are authentic | |
| Precondition: | Sent encrypted ERs | |
| Postcondition: | Verified Hash Value | |
| Basic Flow: | Actor:  1. Poll watchers ensure that no one will intervene the transmission process. | System Response:  1.1 The servers will check the ER’s integrity by looking at the hash code.  1.2 Servers will now include the data in the official count of the votes if the hash codes matches.  1.3 In the event that there was an anomaly in the hash, the servers won’t accept the data thus calling the attention of the administrators. |
| Exceptions: | The machines being used have defects making it impossible to communicate.  Human intervention is present and the system fails to serve its purpose. | |

Activity Diagram

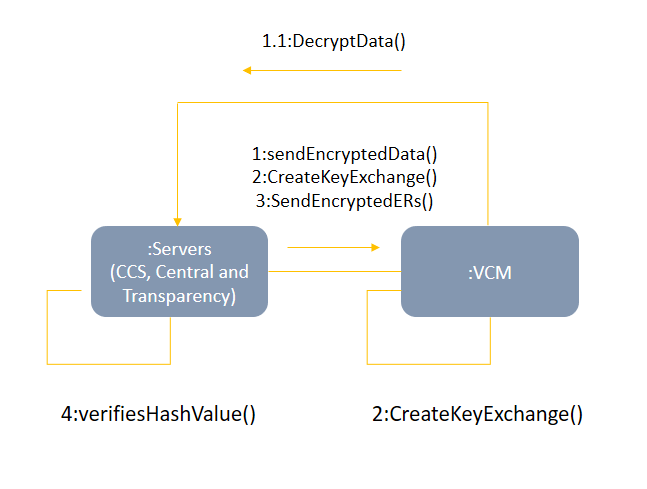


Object Diagram

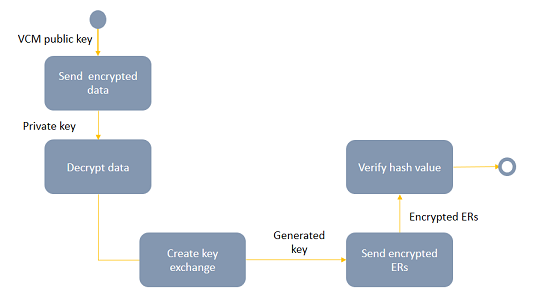


Class Diagram

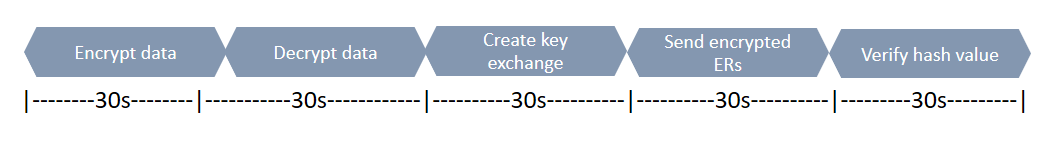
Communication Diagram



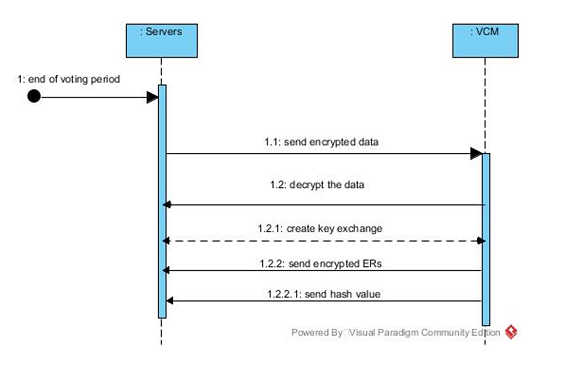
State Diagram



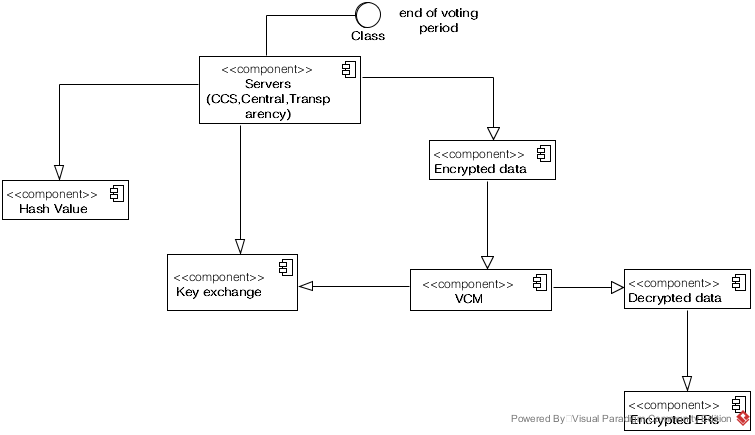
Timing Diagram



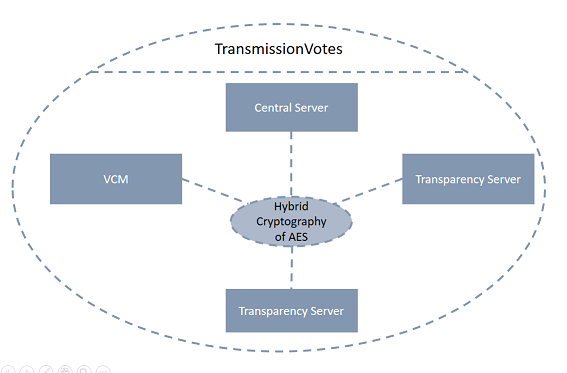
Sequence Diagram



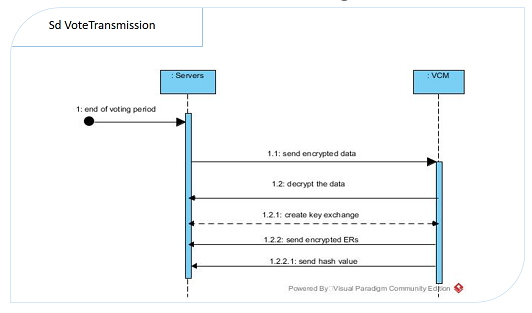
Component Diagram



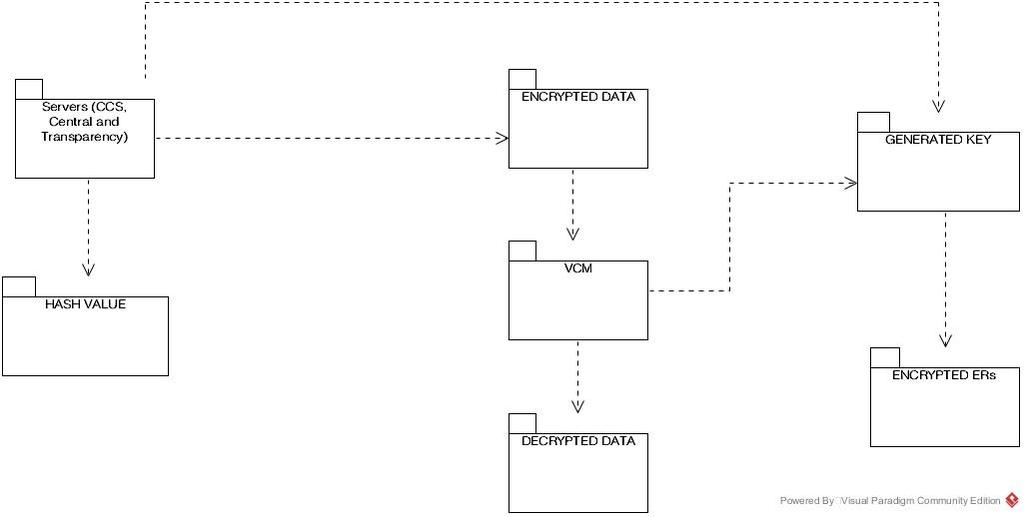
Composite Diagram



Interaction Overview Diagram



Package Diagram



Deployment Diagram

