International Journal of Multidisciplinary Approach and Studies

ISSN NO:: 2348 – 537X

Cryptographic Voting System for Ballot Secrecy And Verifiability

Kristine T. Soberano

Faculty, Northern Negros State College of Science and Technology

ABSTRACT

Observing history, we infer that manual voting system had just a problem with security and results validity. The paradigm shift from hand-based system to paper-based system is caused due to population growth whereas, now, time and safety are so important that it has driven from paper to electronic. There is no dependable reason to stick with manual voting, but there are many security reasons to encourage the use of computerized voting system in order to draw up manual systems to digital era.

This study described the development of an electronic voting system for student council election that has the capability of encrypting data. It determined the quality of the developed system in terms of functionality based on the standard criteria set in McCall's Software Quality Model. It also determined the acceptability and the satisfaction level of college students who used the said system. Through cryptography, ballot verifiability, audit-ability and secrecy had significantly gone better.

Keywords: *Electronic voting, Cryptography, Verifiability, Ballot secrecy, Audit-ability*

INTRODUCTION

One of the great challenges of all democracies is to get their citizens involved in the political system. Obviously, the school is an important arena for equipping younger people with both knowledge and engagement that could benefit the political system. Mock elections is a frequently used pedagogical tool to make the teaching about politics more interesting, and the effect is expected to be increased political awareness and competence among the pupils, and a higher turnout among the first time voters in real elections(nsd.uib.no,2017). Considering that holding elections is a learning process for the development of talents and potentials in students which is a very serious exercise, still issues regarding security, accuracy and integrity of data could not be avoided. Computerized election process has a lot of risks when it comes to security. Systems are vulnerable to tampering and errors, there is little assurance that your vote will count, and voter-verifiable capability feature is not evident for most student council elections. To solve these problems, the researcher developed a system that can:

a. encrypt databases using codes and ciphers,
b. allow public audit,
c. let the voter verify if the vote was counted as cast.

THEORETICAL UNDERPINNING

Student government and elections has evolved differently in different schools. In no place do student governments mean that students are running the school, but there are aspects where students can and do participate and contribute to decision making. Applying democratic principles will result in better schools. When elections are fair and well run according to standards known by all involved then problems are avoided and outcomes are indisputable (School Elections-Note for Students, 2009). In this manner, the development of Crytographic Voting System for Ballot Secrecy and Verifiability would contribute to the democratic practice of having a fair and square student elections.

RELATED LITERATURE

Anonymizing and Aggregating the Ballots

Once a ballot has been correctly encrypted, it is posted on a bulletin board for all to see. The ballots must then be anonymized and aggregated in some way. The process for anonymizing and aggregating ballots differs significantly between the ballot-preserving and aggregate voting systems (David Chaum, 2004).

Aggregate Voting Systems

In the case of aggregate voting systems, the encrypted votes $\{c_j\}_j \epsilon [1,N]$ are generally combined into a single set of cipher-texts $C^{(k)}_{tally}$, where $C^{(k)}_{tally}$ encodes the tallies for race R_k. Depending on the scheme and the size of the election, $C_{(k)tally}$ may be a *sequence* of cipher-texts, though usually never more than one per candidate (i.e. not one per voter, as that would make it a ballot-preserving scheme). Aggregate voting systems typically use homomorphic cryptosystems to maintain and increment aggregate tallies under the covers of encryption. Purposely, practical additive homomorphic scheme were presented in an e.g. yes-vote which is an encrypted '1', a no-vote is an encrypted '0'. The tally is achieved by homomorphic addition, which anyone can do using only the public key, and threshold decryption, which the officials perform together (Josh Benaloh, 1986).

The Voter-Verified Paper Audit Trail

As previously described, one proposed solution for verifiability is the Voter-Verified Paper Audit Trail (VVPAT). Though there are some concerns regarding the practicality of VVPAT machines, there is no question that a properly operating VVPAT machine would significantly simplify the verification chain. VVPAT effectively short-circuits the voting equipment: voters get the ease-of-use associated with computer voting, while the paper trail provides a direct mechanism for verification of the voting machine's output (Josh Benaloh, 1986). However, it is worth noting that even VVPAT does not change the nature of the verification process: a chain of custody, albeit a shorter one, must still be maintained and audited, so that the following questions can be answered:

1. Do the accepted paper trails get properly deposited in the ballot box? Do the rejected paper trails get properly discarded?

2. Are the ballot boxes of paper trails appropriately safeguarded during Election Day?

3. Are the ballot boxes of paper trails appropriately collected and safeguarded after the polls close? What are the safeguards against the introduction of extraneous ballot boxes?

4. Are the paper trails properly tallied? Using what process?

In other words, the VVPAT short-circuits the custody chain prior to ballot casting. The verification process for everything that follows the ballot hand-off, however, remains a chain of custody that must be properly enforced at all times.

End-to-End Verifiability

When dealing with complex systems, software engineering has long relied on the "end-to-end" principle, where, in order to keep the system simple, the "smarts" of the system are kept at higher levels of abstraction, rather than buried deep in the stack (Jerome H. Saltzer, 1984). For example, when routing packets on the Internet, very few assumptions are made about the underlying transport mechanism. Instead, checksums are performed by sender and recipient to ensure end-to-end intergity in the face of random mistakes, and digital signatures are applied to prevent malicious modifications of the data. No details of traffic routing need to be verified in either case; instead, a certain property is preserved from start to finish, regardless of what happens in between (Jerome H. Saltzer, 1984).

Though not all systems are amenable to such a design, voting systems are. Rather than completely auditing a voting machine's code and ensuring that the voting machine is truly running the code in question, end-to-end voting verification checks the voting machine's output only. Rather than maintain a strict chain-of-custody record of all ballot boxes, end-too-end voting checks tally correctness using mathematical proofs. Thus, the physical chain of custody is replaced by a mathematical proof of end-to-end behavior. Instead of verifying the voting equipment, end-to-end voting verifies the voting results (Ronald L. Rivest, 2004).

A Bulletin Board of Votes

Cryptographic voting protocols revolve around a central, digital bulletin board. As its name implies, the bulletin board is public and visible to all, via, for example, phone and web interfaces. All messages posted to the bulletin board are authenticated, and it is assumed that any data written to the bulletin board cannot be erased or tampered with. In practice, implementing such a bulletin board is one of the more challenging engineering aspects of cryptographic voting, as one must worry about availability issues beyond data corruption, such as denial-of-service attacks for both data publication and access. There are, however, known solutions to this problem (Yehuda Lindell, 2002).

Threshold Public-Key Cryptosystems

In many applications, including notably voting, it is desirable to allow decryption only when a quorum of "trustees" agrees. In other words, the secret key *sk* isn't available to a single party. Instead, *l* trustees share *sk*: trustee *i* has share $sk^{(i)}$. If at least *k* of the *l* trustees participates, then decryption is enabled. If fewer than *k* trustees participate, then the security properties of the cryptosystem are fully preserved. There are two conceivable approaches to generating the shares $\{sk^{(i)}\}$. The simpler approach is for a "dealer" to generate (*pk, sk*) normally, split *sk* into shares, then distribute these shares to the appropriate trustees. A more secure approach is to have the trustees generate the key-pair together, with no single party ever learning the complete *sk* in the process (Rivest & Adleman, 1977).

METHODOLOGY

This study took place in Northern Negros State College of Science and Technology, Sagay City Negros Occidental were the participants are shown in the table below:

Table 1. Summary of Respondents of the Study Categorized by Academic Programs

Program	Population	Sample Proportion	Sample Size
BSIT	927	0.086	81
BSF	74	0.086	6
BSED	302	0.086	27
BSBIO	81	0.086	6
BSHRM	312	0.086	28
BSCRIM	1305	0.086	112
BSBA	870	0.086	75
ABE	364	0.086	31
TOTAL POPULATION	**4,235**		**366**

The researcher used the stratified proportional random sampling technique to specifically measure operational efficiency, effectiveness and user satisfaction of the system. Stratified proportional random sampling is used to provide valid and accurate result using Slovin's formula of getting the sample size dividing the whole population of respondents into smaller groups called strata.

Research Instrument

Research Instruments depends and varies on the nature of the problem. In this study, there were three (3) survey questionnaires that were designed and distributed to the respondents. During the needs assessment, the researcher crafted a self-made instrument validated by three (3) IT experts using Carter V. Good and Douglas F. Scates evaluation of instruments. For the system/software evaluation during initial and final testing, the instrument used was the standard questionnaire which is the McCall's Software Quality Model. Meanwhile, for the evaluation of the implementation of the system to determine its efficiency, effectiveness and user satisfaction, the researcher used Criteria- based Assessment which is the ISO/IEC 1991, Nielsen's Usability Engineering, 1993 and Lund, A.M., 2001 USE Questionnaires.

Data Analysis Procedures

The researcher gathered and collected the data in order to tabulate its statistical interpretation. The mean was used to determine software quality, efficiency, and user satisfaction.

Software Design and Development Component

Software Design

The Software development model used was the modified waterfall approach. It includes several stages such as Requirements analysis, System and Software Design, Implementation and Unit Testing, Integration and System Testing and Operation, and Maintenance as shown in Figure 1.

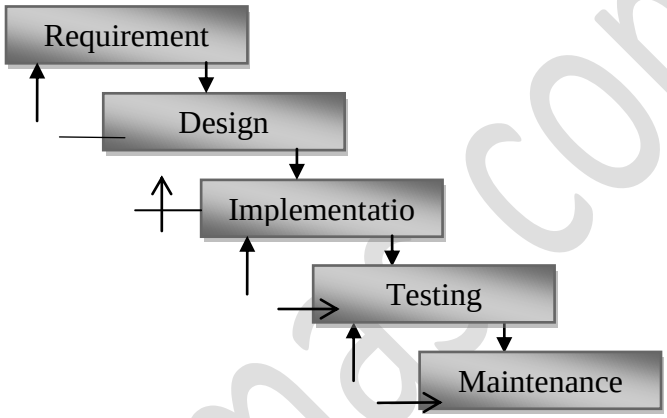

Figure 1. Modified Waterfall Approach

During the requirement phase, the researcher gathered all the needed data such as student profile, the programming language to be used in encryption and decryption, and other necessary data used in election process. On the design phase, simple fonts, buttons, forms, and interfaces were created to ensure a user-friendly navigation. Coding also took place on this phase. The Cryptographic Voting System was initially installed for the purpose of evaluation. Upon the recommendation of the IT experts, the bugs and other errors of the system were fixed to meet the full specifications.

The voter experience was also simple and mostly familiar:

1. **Sign in**. The students will be given a ticket with encrypted password for them to log in the system. Once the system accepts the password, the voter can now start voting. The ticket can only be used once. If the student casts her/his vote the password will immediately be deleted from the database to make sure that it cannot be used twice. Due to a large number of voters of in the main campus, the researcher chose to create a system generated password.

2. **Make selection**. The students will just simply click the name and picture of their selected candidates.

3. **Casting**. After the selection, the candidate must click “cast vote” button so that their votes will be counted. The session will automatically expire a few seconds after the vote is cast.

4. **Verification**. Any voter can verify his vote because the system provides verification feature.

5. **Public Audit**. The system has a partial tally feature that counts the number of votes during the ongoing election process. The partial tally result can be viewed and monitored outside the precinct thru digital bulletin board.

RESULTS AND DISCUSSION

The data concerning cryptosystem can reconcile public audit-ability and ballot secrecy in voting performance of the college students. The implementation of the system and its relationship with operational efficiency to organized truthful performance were presented, analyzed, and interpreted.

Table 2.0 The Extent of Operational Efficiency of Cryptographic Voting System For Ballot Secrecy and Verifiability based on McCall's Software Quality Model

Criteria	Mean	Interpretation
Audit-ability The ease with which conformance to standards can be checked.	4.54	Very Good
Accuracy The precision of computations and control	4.57	Very Good
Completeness The degree to which full implementation of the required functions has been achieved.	4.69	Very Good
Communication Commonality The degree to which standards interfaces and protocols are understood.	4.78	Very Good
Conciseness The compactness of the program in terms of lines and code.	4.77	Very Good
Consistency The use of uniform design and documentation techniques throughout the software development project.	4.80	Very Good
Observability The process of streaming the software components can be easily identified and understand.	4.88	Very Good
Operability The ease of operation of the program.	4.62	Very Good
Security The availability of mechanisms that control or protect programs and data.	4.51	Very Good
Self-Documentation The degree to which the source code provides meaningful documentation.	4.53	Very Good
Simplicity	4.80	Very Good

Criteria	Mean	Interpretation
The degree to which the program can be understood without difficulty.		
Software System Independence The degree to which the program is independent of non-standard programming language features, operating system characteristics, and other environmental constraints.	4.63	Very Good
Traceability The ability to trace a design representation or actual program component back to requirements.	4.80	Very Good
Training The degree to which the software assists in enabling new users to apply the system.	4.89	Very Good
Controllability The system can be easily controlled and manipulated in terms of execution, program structure, and design.	4.70	Very Good
Data Commonality The use of standard data structures and types throughout the program.	4.72	Very Good
Decomposability The software is built from series of modules, and can be tested independently.	4.90	Very Good
Error Tolerance The damage that occurs when the program encounters an error.	4.67	Very Good
Execution Efficiency The run-time performance of the program.	4.80	Very Good
Expandability The degree to which architectural, data, or procedural design can be extended.	4.83	Very Good
Generality The breadth of potential application of program components.	4.80	Very Good
Hardware Independence The degree to which the software is decoupled from the hardware on which it operates.	4.69	Very Good
Instrumentation The degree to which the program monitors its own operation and identifies errors that do occur.	4.77	Very Good
Modularity The functional independence of program components.	4.60	Very Good
Total	**4.35**	**Very Good**

Rating Scale: [5] Very Good [4] Good [3] Average [2] Poor [1] Very Poor

Table 2 reveals the efficiency of the developed system based on the ratings given by the experts. Overall, the system was rated 4.35 which was interpreted as Very Good.

Table 3. Level of Users Satisfaction on Cryptographic Voting System for Ballot Secrecy and Verifiability

Scale: 5 – Very Satisfied 4 – Satisfied 3 – Moderately Satisfied 2 – Poorly Satisfied 1 – Not Satisfied	**Mean**	**Interpretation**
It allows the voter to verify his/her vote.	4.28	Very Satisfied
It saves me time when I use it	4.52	Very Satisfied
It does everything I would expect it to do	4.74	Very Satisfied
It is easy to use	4.38	Very Satisfied
It requires the fewest steps possible to accomplish what I want to do with it	4.47	Very Satisfied
It has the ability to publicly display the total number of votes	4.30	Very Satisfied
I can use it without written instructions	4.29	Very Satisfied
I don't notice any inconsistencies as I use it	4.54	Very Satisfied
I can recover from mistakes quickly and easily	4.37	Very Satisfied
I learned to use it quickly	4.69	Very Satisfied
I easily remember how to use it	4.75	Very Satisfied
I am satisfied with it	4.47	Very Satisfied
It is fun to use	4.58	Very Satisfied
It is pleasant to use	4.57	Very Satisfied
If there are "help" and "hint" messages, they are easy to access	4.46	Very Satisfied
The program tolerates variations in command formats (e.g. upper or lower case, extra space, etc.)	4.27	Very Satisfied
Computer capabilities such as graphics, color, text and buttons are used appropriately	4.30	Very Satisfied
Printouts are clear and well organized	4.51	Very Satisfied
Updates can be loaded easily into the system	4.75	Very Satisfied
Records are kept secured and confidential	4.55	Very Satisfied
Over all, I am satisfied with the system	4.78	Very Satisfied
TOTAL	**4.50**	**Very Satisfied**

The data on the table showed that the students of Northern Negros State College of Science and Technology were very satisfied in using Cryptographic Voting System For Ballot Secrecy and Verifiability by giving a mean of **4.50** with a very satisfied interpretation.

CONCLUSION

This unswerving system was fully tested and the efficiency aspect of the voting processes was very efficient. Voters were able to vote electronically in a timely fashion and with minimal directions. Similarly, encryption code were observed and evaluated by the researcher and the experts to draw up cryptosystem performance in order not to jeopardize the ballot verifiability and secrecy.

RECOMMENDATIONS

On the basis of the conclusion, the following recommendations are the following:

1. Schools other than Northern Negros State College of Science and Technology may consider using Cryptographic Voting System for Ballot Secrecy and Verifiability to ensure data security, voter-verifiability and audit-ability.

2. Further studies may be conducted. Perhaps, making the voting system online.

REFERENCES:

i. Backes, M. and Maffei, M.(2008). Automated Verification of Remote Electronic Voting Protocols in the Applied Pi-calculus.,Computer Security Foundations Symposium.

ii. Chaum, David.(2004). Secret-Ballot Receipts: True Voter-Verifiable Elections. Security and Privacy.

iii. Gritzalis, Dimitris.(2002).Secure Electronic Voting. Kluwer Academic Publishers.

iv. Jones, Douglas W. (2001-2003). A Brief Illustrated History of Voting.

v. Neff, Andrew C.(2006). Assisted Human Interactive Proofs: A Formal Treatment of Secret Voter Receipts.

vi. Rubin,Avi. (October 2004).An Election Day Clouded by Doubt, retrieved from http://avirubin. com/vote/op-ed.html.

vii. Shapiro, Ari.(October2004) Absentee Ballots Go Missing in Florida's Broward County.

viii. Yehuda, Lindell.(2002). On the Composition of Authenticated Byzantine Agreement.