A COMPARATIVE STUDY OF GENERIC CRYPTOGRAPHIC MODELS FOR SECURE ELECTRONIC VOTING

Okediran O. O.

Department of Computer Science & Engineering,
Ladoke Akintola University of Technology, P.M. B. 4000, Ogbomoso, Nigeria

Omidiora E. O.

Department of Computer Science & Engineering,
Ladoke Akintola University of Technology, P.M. B. 4000, Ogbomoso, Nigeria

Olabiyisi S. O.

Department of Computer Science & Engineering,
Ladoke Akintola University of Technology, P.M. B. 4000, Ogbomoso, Nigeria

Ganiyu R. A.

Department of Computer Science & Engineering,
Ladoke Akintola University of Technology, P.M. B. 4000, Ogbomoso, Nigeria

ABSTRACT

Electronic voting has been attracting considerable attention during the last years. The interest in e-voting is based on one hand upon interest and attention devoted to e-government, e-democracy, e-governance, etc. On the other hand, interest in e-voting is founded in problems with domestic election systems, e.g. violence, intimidation, ballot stuffing, under-age and multiple voting, counting error, complicity of the security agencies and the absence or late arrival of election materials lacking e.t.c. However, without appropriate security measures, electronic based elections can be a challenge. This paper reviews four generic cryptographic models that were proposed in the academic literature for secure electronic voting and provides a comparison amongst the four models in terms of their core properties of universal verifiability, support for write-in ballot, efficient voting, efficient tallying and large scale election support.

KEYWORDS: e-voting, election, homomorphic model, mix-net model, blind signature model, verifiable secret sharing model

1. INTRODUCTION

Elections allow the populace to choose their representatives and express their preferences for how they will be governed. Naturally, the integrity of the election process is fundamental to the integrity of democracy itself. The election system must be sufficiently robust to withstand a variety of fraudulent behaviors and must be sufficiently transparent and comprehensible that voters and candidates can accept the results of an election (Kohno *et al.*, 2003). Correctness, robustness to fraudulent behaviors, coherence, consistency, security, and transparency of voting are all key requirements for the integrity of an election process (Malkawi *et al.*, 2009).

There is a wide variety of different voting systems that are based on traditional paper ballots, mechanical devices, or electronic ballots (NSF, 2001). In a traditional paper ballots, voters choose or mark their favourite choices on ballots and place them in boxes, which are sealed and officially opened under special conditions to warrant transparency. The ballots are then counted manually, which is a tedious process that is subject to human error. With voting via mechanical systems, meanwhile, voters make their choices by pulling down on mechanical levers that correspond to their favourite choice of candidates. Each lever has a mechanical counter that reports the number of votes for that position. These machines are no longer manufactured (NSF, 2001). On the other hand, some systems use punch cards where voters punch holes in computer readable ballot cards. These systems are not reliable because of problems in reading cards and were replaced by optical scan device systems, which allow voters to record choices by filling in areas on the ballots. The ballots are read using a computer scanner and then the votes are counted automatically using a computer program (NSF, 2001). Finally, special-purpose computers are used as voting machines where voters use touch screens or push buttons to select choices, which are stored and counted or processed by a special program on the same machine (NSF, 2001). Often times, however, counting errors take place, and in some cases, voters find ways to vote more than once, introducing irregularities in the final count results, which could, in rare cases, require a repeat of the election process altogether. Moreover, in some countries, purposely introduced manipulations of the votes take place to distort the results of an election in favour of certain candidates. Although such mishaps can be avoided with a properly scrutinized election process, errors can still occur, especially when the number of voters is quite large. Quite often, international monitoring bodies in certain countries.

The advancement of information and telecommunications technologies allow for a fully automated online computerized election process. In addition to overcoming commonly encountered election pitfalls, electoral vote counts are done in real time that by the end of elections day, the results are automatically out (Mercuri, 2000; Rubin, 2002). The election process can be easily enhanced with various features based on the demand and requirements of different countries around the world. E-voting is an interdisciplinary subject and should be studied together with the experts of different domains, such as software engineering, cryptography, politics, law, economics and social sciences. Although many people have worked on this subject, mostly e-voting is known as a challenging topic in cryptography because of the need to achieve voter anonymity and therefore, to ensure his/her privacy (Cetinkaya and Cetinkaya, 2007).

Furthermore, critics of e-voting claim that the technology is not mature enough for protecting voter privacy, securely authenticating online voters, and for ensuring the integrity of the voting and tallying stages in a universally verifiable way. Moreover, there is the fear that the digital divide will skew political power towards non-minorities (Rubin, 2002; Kohno *et al.*, 2003). In general, e-voting systems are expected to satisfy the following security goals (Neumann, 1993; Benaloh and Tuinstra, 1994; Cranor and Cytron, 1997):

i. *Democracy:* All eligible voters must be able to vote, one person - one vote and no one can vote more than once or vote for others;

ii. *Accuracy:* votes cannot be altered, duplicated or eliminated from the final tally;
iii. *Privacy:* After casting a vote, no one should be able to link the voter to this vote;
iv. *Fairness:* all votes remain secret while the voting period is not completed;
v. *Verifiability:* Voters can independently verify that their votes have been counted correctly and are included in the final tally;
vi. *Reliability:* Election systems should work robustly, without loss of any votes, even in the face of numerous failures, including failures of voting machines and total loss of Internet communication;
vii. *Receipt-freeness:* no voter should be able to prove to others how he/she voted (even if he/she wants to);
viii. *Uncoercibility:* no party should be able to coerce a voter into revealing his/her vote.

Cryptography is naturally used to secure transactions in complex systems where the interests of the participating entities are in conflict. Not surprisingly, cryptography is one of the most significant tools for securing online voting protocols. While in traditional elections most ideal security goals such as democracy, privacy, accuracy, fairness and verifiability, are assured to a point given physical and administrative premises, this same task is quite difficult in online elections. For example, receipt-freeness and verifiability seem to be contradictory: when voting over the Internet, the very means that allow a voter to verify that his/her vote was counted properly (e.g. receipts, vote encrypting keys, user-selected randomness, etc), may also allow a dishonest third party to force the voter to reveal his/her vote. Another controversial pair of security properties are privacy and eligibility: it seems difficult in online elections to unequivocally identify and check the credentials of a voter, while at the same time protecting the privacy of his/her vote.

In the following sections we reviewed four proposed generic cryptographic models in academic literature for secure electronic voting and carried out a comparison amongst the four models in terms of their core properties of universal verifiability, support for write-in ballot, efficient voting, efficient tallying and large scale election support.

2. Cryptographic Models for Secure Electronic Voting

Since the first cryptographic protocols for electronic elections was published (Chaum, 1981; Demillo *et al.*, 1982; Benaloh, 1987), several solutions have been described in academia to deal with the security problems in online voting. In this section we review the generic models and assess their suitability in terms of the following criteria: universal verifiability, support for write-in ballots, efficient voting, efficient tallying and large-scale support.

2.1 The Mix-net Model

Mix networks (mix-nets), introduced in (Chaum,1981), usually consist of a set of servers (mixes) which accept a batch of input messages and output the batch in randomly permuted (mixed) order so that the input and output messages are unlinkable. Figure 1 depicts then general case of voting with mix-net model.

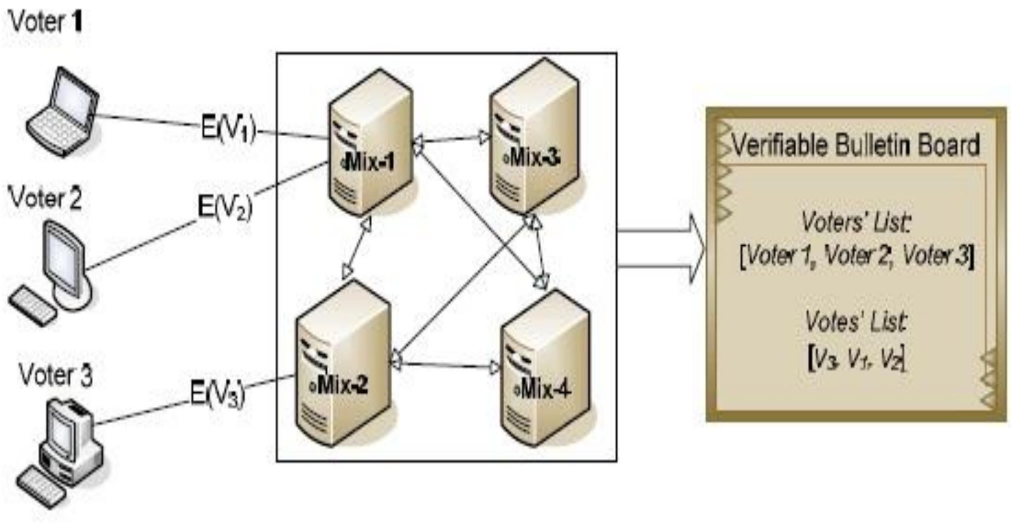

Figure 1: The General Case of Voting with mix-net

Although originally proposed for anonymous e-mail communication between distrusting entities, mix-nets in online elections aim at hiding the origin of a ballot: tallying officials permute and randomize the encrypted ballots so that the link between the identity of the voter and the vote is broken. Depending on the mixing mechanism, mix-nets can be classified into re-encryption mix-nets and decryption mix-nets.

2.1.1 Re-encryption mix-net

This type of mix-net (Ogata *et al.*, 1997; Jakobsson, 1999; Golle *et al.*, 2004; Nguyen *et al.*, 2004) usually relies on a public key cryptosystem, which allows re-encryption of the input messages with a random number. Most re-encryption mix-nets use randomized public-key encryption schemes such as the (ElGamal, 1985) or the (Paillier, 1999) cryptosystems, where the size of the cipher texts can be independent of the number of the involved mix servers. In a typical implementation, individual votes are encrypted with the public key of the mix-net, while the decryption key is shared among the mix servers. Then the list of encrypted votes is sequentially re-encrypted and shuffled in each mix server.

The transformations are secret and verifiable, even if a number of mix servers are malicious. The final list of encrypted votes is decrypted by a number of honest mix servers, using threshold decryption techniques (Desmedt, 1994). A few cryptographic schemes for example (Hirt and Sako, 2000; Neff, 2001; Juels *et al.*, 2002; Jakobsson *et al.*,2002; Acquisti, 2004; Aditya *et al.*,2004) employ re-encryption mix-nets to protect voter privacy, since this model adds flexibility by separating the mixing and the decryption phases. A typical re-encryption mix-net for voting is shown in Figure 2.

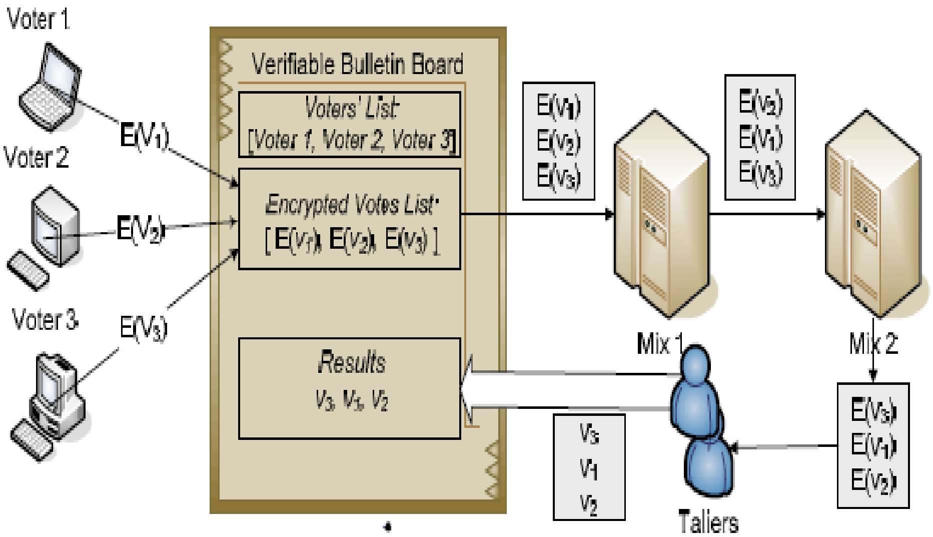

Figure 2: Re-encryption-mix net model

2.1.2 Shuffle decryption mix-net

This type of mix-net (Chaum, 1981; Abe, 1998; Furukawa, 2004) accepts as input a collection of cipher texts and outputs the corresponding list of plaintexts in a randomly permuted order. A number of independent mixers sequentially perform the shuffling and decryption of encrypted votes in a way that the final votes cannot be linked to the original set of encrypted votes, while at the same time the verifiability of the correct output is established. Shuffle decryption is considered as more efficient than re-encryption shuffles.

However in case of failure of one of the mix servers, systems based on shuffle decryption usually need more computation to recover (Furukawa, 2004). The mix-net model (both re-encryption and shuffle decryption techniques) satisfies voter privacy, verifiability, and robustness. In the optimum scenario, voter privacy is assured if at least one mix server behaves honestly and does not reveal the relation between its input and output links. To satisfy the verifiability criterion, the servers must prove or at least provide strong evidence e.g. (Jakobsson *et al.*, 2002) that their shuffles were correctly constructed; otherwise a malicious server could insert fake votes to the final tally. These proofs are constructed using zero-knowledge techniques (Goldreich *et al.*, 1991), so that no information is provided about the secret shuffle, besides that the shuffle was correct. In universally verifiable mix-nets e.g. (Abe, 1998), an independent observer is able to verify that the output of each mix was correctly computed from the input. Alternatively, the servers may establish verifiability among them and then validate the generated list e.g. (Jakobsson, 1999).

Mixnet elections require fewer interactions by the voters and have inherent support for “write in” ballots. A disadvantage of mix-nets is that in their fully robust form they may need complex protocols for generating and maintaining shared private keys, as well as for mixing and proving correctness of the shuffles. Mix-nets can be efficient if:

i. the computation required by a voter is independent of the number of mix servers;

ii. the complexity involved at the server-side processing can be tolerable; and
iii. the verifiability checks can be kept substantially low.

Recent results, have improved the efficiency and practicality of mix-nets e.g. (Furukawa, 2004; Nguyen *et al.*, 2004).

2.2 The Homomorphic Model

According to this model, introduced in (Cramer *et al.*, 1997) and extended in (Baudron *et al.*, 2001), each voter signs and publishes an encryption of his/her vote. Encrypted votes are then "added" into the final tally, to form an encryption of the "sum" of the submitted votes. The model is based on the algebraic homomorphic properties of several probabilistic public key cryptosystems. These cryptosystems encrypt a message M by raising a base g to the power M modulo a large prime number, and then randomizing the result. With homomorphic encryption there is an operation $\oplus$ defined on the message space and an operation $\otimes$ defined on the cipher space, such that the "product" of the encryptions of any two votes is the encryption of the "sum" of the votes, i.e.:

$$EM_1 \oplus EM_2 = E\,(M_1 \otimes M_2)$$

This property allows either to tally votes as aggregates or to combine shares of votes (Benaloh, 1987; Schoenmakers, 1999), without decrypting single votes. However, each vote must belong to a well-determined set of possible votes such as {+1, -1} for {"yes", "no"} votes. Moreover, each voter must provide a universally verifiable proof that his/her vote belongs to the predefined set of votes, otherwise, it would be easy for a malicious voter to manipulate the final tally.

After the voting period has closed, a threshold of election authorities cooperatively decrypts the final tally. The results are published on a bulletin board and the accuracy of the voting stage is verified. Depending on the level of trust given to them, the authorities may also provide a publicly verifiable proof that the decryption was correct. In this way individual voters and/or external observers can be assured that all the votes were counted correctly. An example of the homomorphic voting model is shown in Figure 3.

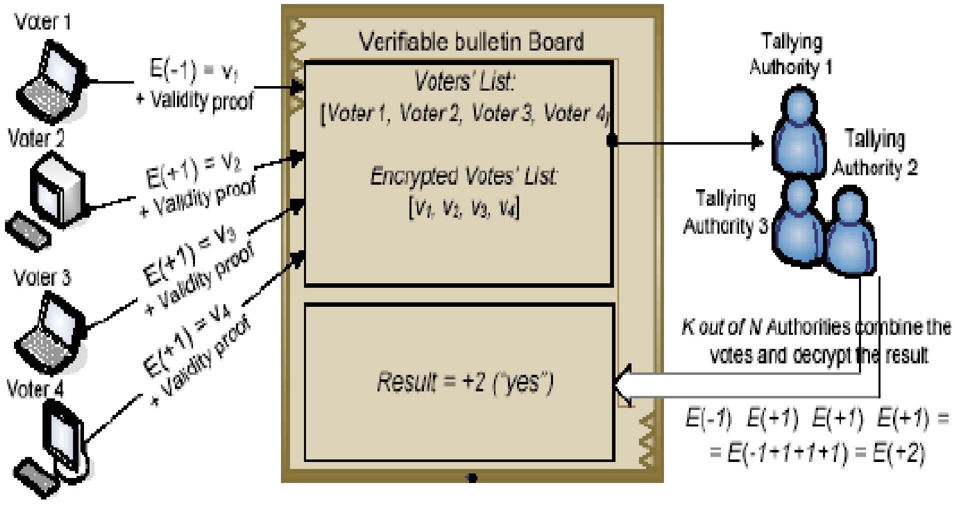

Figure 3: The homomorphic model (Cramer *et al.*, 1997)

While the original model provides a general framework that allows usage of any probabilistic encryption scheme, only few probabilistic encryption schemes can scale well in large elections with multiple candidates. For example, in (Cramer *et al.*, 1997) a variant of the ElGamal encryption scheme required an exhaustive search over all possible election results by the authorities for the computation of the final tally. Recent proposals have been based on additively homomorphic public key cryptosystems with trapdoor decryption of discrete logarithms (Paillier, 1999; Baudron *et al.*, 2001; Damgard *et al.*, 2003), in order to allow handling of very large tallies.

The homomorphic model satisfies the accuracy, privacy, fairness, robustness and universal verifiability properties. It also inherently supports prevention of double voting, since the voters do not need to be anonymous. It works well in elections where ballots have only questions of a K-out-of-L type, which precludes write-in ballots. Another unattractive feature is that voters may need to run special-purpose code on their computer, for constructing the zero-knowledge proof of validity for their vote.

2.3 The Blind Signature Model

Election protocols of this category, introduced in (Fujioka *et al.*, 1992), enable voters to get their vote validated from an election authority, while preserving the secrecy of their vote. Blind signatures (Chaum, 1982) are the electronic equivalent of signing carbon-paper-lined envelopes: a user seals a slip of a paper inside such an envelope, and later gets it signed on the outside. When the envelope is opened, the slip will bear the carbon image of the signature. When used in an online voting protocol, a voter encrypts, then blinds the vote, and presents it to a validating authority for validation. After the authority validates the vote, the voter un-blinds the encrypted vote and gets a validated vote that cannot longer be correlated to the original blinded message. The voter then uses an anonymous channel to submit the validated vote to the tallying authorities, as shown in Figure 4.

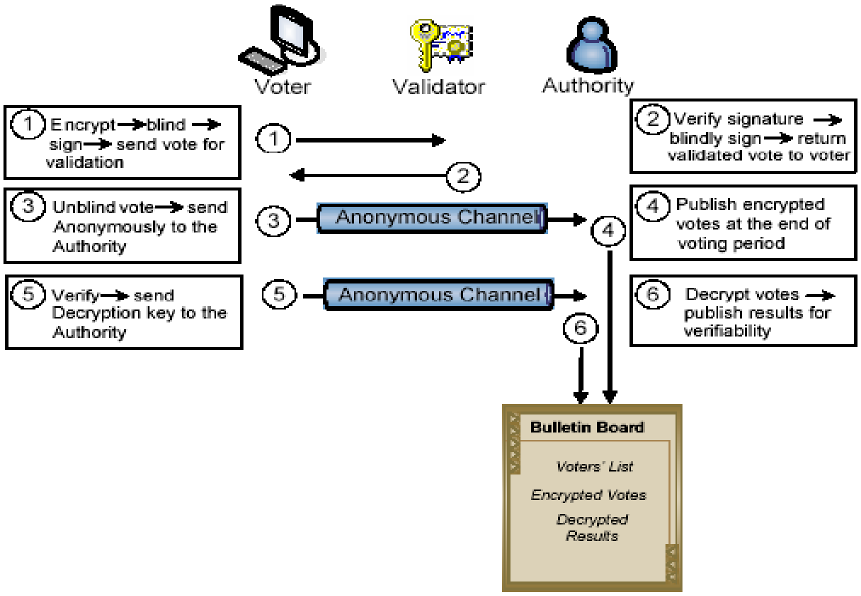
Voter
Validator
Authority
1 Encrypt → blind → sign → send vote for validation
2 Verify signature → blindly sign → return validated vote to voter
1
2
3 Unblind vote → send Anonymously to the Authority
3
Anonymous Channel
4
4 Publish encrypted votes at the end of voting period
5 Verify → send Decryption key to the Authority
5
Anonymous Channel
6
6 Decrypt votes → publish results for verifiability
Bulletin Board
Voters' List
Encrypted Votes
Decrypted Results

Figure 4: Blind Signature Model (Fujioka *et al.*, 1992)

Protocols within this model are simple, easily manageable, computationally efficient and naturally support "write-in" ballots. A problem with early schemes (Fujioka *et al.*, 1992; Cranor and Cytron, 1997; Herschberg, 1997) was the ability of a malicious server to impersonate absentee voters in the final tally, thus violating the democracy criterion. In the original model (Fujioka *et al.*, 1992) two-phase voting was supported to achieve fairness: voters submitted their encrypted vote and then waited until the end of the election to submit their vote-opening keys. In (Cranor and Cytron, 1997; Herschberg, 1997) the protocol of (Fujioka *et al.*, 1992) was changed to allow voters to vote and walk away, however in both protocols there is the risk that a malicious authority learns intermediate results, therefore violating the fairness property. In subsequent proposals (Ohkubo *et al.*, 1999; Durette, 1999; Joaquim *et al.*,2003; Lebre *et al.*, 2004) the power of administration is distributed among multiple authorities so that:

i. no election administrator is able to impersonate legitimate voters in the final tally, and
ii. the results are becoming available only at the end of the election.

To establish robustness in the election process, threshold techniques were also proposed (Ohkubo *et al.*,1999; Joaquim *et al.*, 2003; Lebre *et al.*,2004). For example, in (Ohkubo *et al.*, 1999), a (t, N) threshold cryptosystem assured that as long as $N\text{-}t+1$ counters are honest, the results will only be available at the end of the election.

2.4 The Verifiable Secret Sharing Model

This model (Benaloh, 1987) uses a homomorphic secret sharing scheme. With such schemes there is an operation $\oplus$ defined on the share space, such that the "sum" of the shares of any two secrets x_1 , x_2 is a share of the secret $x_1 \oplus x_2$. In the voting scheme proposed in (Benaloh, 1987) each voter shares his/her vote among n voting authorities. The shares are encrypted with the public key of the receiving authority, authenticated, and posted on a bulletin board. At the end of the voting period each authority adds all the received shares to get an encrypted share of the tally. Finally the authorities combine their shares to get the encrypted tally. Thus no single vote is ever decrypted. For robustness, a (t, N) homomorphic threshold scheme is used: then only t out of N authorities needs to combine their (true) shares. Late schemes employ this model in a universally verifiable way, both in the sharing and tallying phases (Cramer *et al.*, 1996; Schoenmakers, 1999). The verifiable secret sharing model achieves voter privacy, robustness and universal verifiability. Protection from double voting is analogous to the homomorphic model. In order to prevent voters from disrupting the election by sending false shares to authorities, voters similarly need to construct zero knowledge proofs of validity for their votes. Compared with the homomorphic model, verifiable secret sharing moves computation and communication burden from talliers to voters. This method requires communication between a voter and all servers, while the talliers do not need to run a shared-key generation protocol for a threshold decryption scheme. As a result, it can be considered as more suitable for small-scale elections, where voters may be talliers as well.

3. Conclusion

Table 1 summarizes a comparison amongst the four cryptographic models for secured electronic voting reviewed in this paper in terms of their core properties of universal verifiability, support for write-in ballot, efficient voting, efficient tallying and large scale election support. The comparison shows that blind signature model is the most efficient cryptographic model for secure electronic voting as it supports more core properties desirable for secure e-voting than any other model reviewed in this paper.

Table 1: Comparison of Generic Cryptographic models

Properties	**Universal Usability**	**Write-in Ballot**	**Efficient Voting**	**Efficient Voting**	**Large-scale Support**
Models					
Mix-net Model	Yes	Yes	Yes	No	No
Homomorphic Model	Yes	No	No	Yes	Yes
Blind Signature Model	No	Yes	Yes	Yes	Yes
Verifiable Secret Sharing	Yes	No	No	Yes	No

4. References

Abe, M. (1998), “Universally Verifiable Mix-net with Verification Work Independent of the Number of Mix-centers”. In Proceedings of the Advances in Cryptology – EUROCRYPT 98, LNCSVol. 1403. Springer-Verlag, 437–447.

Acquisti A. (2004),”Receipt-Free Homomorphic Elections and Write-In Ballots”. Tech. Rep. 2004/105, CMU-ISRI-04-116, Carnegie Mellon.

Aditya R., Lee B., Boyd C. and Dawson E., (2004),” An Efficient Mixnetbased Voting Scheme Providing Receipt-Freeness”. In Proceedings of the 1st Trustbus 2004, LNCS Vol. 3184. Springer-Verlag, 152–161.

Baudron O., Fouque P., Pointcheval D., Poupard G., and Stern, J. (2001). “Practical Multi-Candidate Election System”. In Proc. of the 20th ACM Symposium on Principles of Distributed Computing. ACM Press, 274–283.

Benaloh J. (1987), “Verifiable Secret Ballot Elections”. Ph.D. Thesis, Yale University.

Benaloh J. and Tuinstra D., (1994), “Receipt-Free Secret-Ballot Elections”. In Proceedings of the 26th Annual ACM Symposium on Theory of Computing. ACM Press, 544–553.

Cetinkaya O. and Cetinkaya D., (2007), “Verification and Validation Issues in Electronic Voting”. The Electronic Journal of e-Government Volume 5 Issue 2, pp 117- 126.

Chaum D. (1981), ”Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms”. Communication of ACM 24, 2, 84–88.

Chaum D. (1982),” Blind Signatures for Untraceable Payments”. In Proceedings of the Advances in Cryptology – CRYPTO’82. Plenum Press, 199–203.

Cramer R., Gennaro R., and Schoenmakers B. (1997), “A Secure and Optimally Efficient Multi-Authority Election Scheme”. European Transaction on Telecommunications 8, 5, 481–490.

Cranor L. and Cytron R., (1997), “Sensus: A Security-Conscious Electronic Polling System for the Internet”. In Proceedings of the International Conference on System Sciences. Wailea, Hawaii.

Damgard I., Jurik M., and Nielsen J., (2003), ”A Generalization of Paillier’s Public-Key System with Applications To Electronic Voting”. International Journal of Information Security to Appear.

Demillo R., Lynch N., and Merritt M., (1982),”Cryptographic Protocols”. In Proceedings of the 14th Annual ACM Symposium on Theory of Computing. ACM, 383–400.

Desmedt Y., (1994), “Threshold Cryptography”. European Transactions on Telecommunications 5, 4, 449–457.

Durette, B. W., (1999), “Multiple Administrators for Electronic Voting”. M.Sc. Thesis, Massachusetts Institute of Technology.

ElGamal T., (1985), “A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms”. IEEE Trans. on Information Theory 30, 4, 469–472.

Fujioka A., Okamoto T., and Ohta K., (1992). “A Practical Secret Voting Scheme for Large Scale Elections”. In Proceedings of the Advances in Cryptology – AUSCRYPT ’92. LNCS, vol. 718. Springer-Verlag, 244–251.

Furukawa J,. (2004), “Efficient, Verifiable Shuffle Decryption and its Requirement of Unlinkability”. In Proceedings of the Public Key Cryptography (PKC’04). LNCS, vol. 2947. Springer- Verlag, 319–332.

Goldreich O., Micali S., and Widgerson A., (1991), ” Proofs That Yield Nothing But Their Validity, or All Languages in NP Have Zero-Knowledge Proof Systems”. Journal of the ACM 38, 691–729.

Golle P., Jakobsson M., Juels A. and Syverson P., (2004), ”Universal Re-Encryption for Mixnets”. In Proceedings of the RSA Conference Cryptographers Track ’04, T. Okamoto, Ed. LNCS, vol. 2964. Springer-Verlag, 163–178.

Herschberg M., (1997), “Secure Electronic Voting Using the World Wide Web”. M.Sc. Thesis, MIT.

Hirt M. and Sako K. (2000), “Efficient Receipt-Free Voting Based on Homomorphic Encryption”. In Proceedings of the Advances in Cryptology-EUROCRYPT’00. LNCS, vol. 1807. Springer-Verlag, 539–556.

Jakobsson M., (1999),” Flash Mixing”. In Proceeding of the PODC’99. IEEE, 83–89.

Jakobsson M., Juels A., and Rivest R., (2002), ”Making Mix Nets Robust for Electronic Voting by Randomized Partial Checking”. In Proceedings of the 11th USENIX Security Symposium. IEEE, 339–353.

Joaquim R., Zuquette, A., and Ferreira P., (2003),” REVS - A Robust Electronic Voting”

Juels A., Catalano D. and Jakobsson M., (2002), “Coercion-Resistant Electronic Elections”. Cryptology ePrint Archive 165.

Kohno T., Stubblefield A., Rubin A. Wallach D. S., (2003), “Analysis of an Electronic Voting System”. Johns Hopkins University Information Security Institute Technical Report TR-2003-19.

Lebre R., Joaquim R., Zqueta A., and Ferreira P., (2004), “Internet Voting: Improving Resistance to Malicious Servers in REVS”. In Proceedings of the International Conference on Applied Computing – IADIS’04.

Malkawi M., Khasawneh M., Al-Jarrah O., (2009), “Modeling and Simulation of a Robust E-voting System. Communications of the IBIMA, Volume 8, 2009. ISSN: 1943-7765.

Mercuri R., (2000), “Electronic Vote Tabulation Checks and Balances”. Ph.D thesis, University of Pennsylvania, Philadelphia.

Neff A. (2001), “A Verifiable Secret Shuffle and Its Application to E-Voting”. In Proceedings of the 8th Computer and Communications Security Conference. ACM, Philadelphia, USA.

Nguyen L., Safavi-Naini, R. and Kurosawa K., (2004), “Verifiable Shuffles: A Formal Model and a Paillier-Based Efficient Construction with Provable Security”. In Proceedings of the ACNS04. LNCS, vol. 3089. Springer-Verlag, 236–247.

Neumann P. G., (1993), “Security Criteria for Electronic Voting”. In Proceedings of the 6th National Computer Security Conference. IEEE.

NSF, (2001). “Report on the National Workshop on Internet Voting: Issues and Research Agenda National Science Foundation, at http://news.findlaw. com /cnn/docs/voting/ nsfevoterprt. pdf.

Ogata W., Kurosaw K., Sako K. and Takatani, K. (1997), “Fault Tolerant Anonymous Channel”. In Proceedings of the 1st International Conference on Information and Communications Security – ICICS. LNCS, vol. 1334. Springer-Verlag, 440–234.

Ohkubo M., Miura F., Abe M., Fujioka A., and Okamoto T., (1999), ”An Improvement on a Practical Secret Voting Scheme”. In Proceedings of the Information Security Conference – IS’99. LNCS, vol. 1729. Springer-Verlag, 225–234.

Paillier P., (1999),”Public Key Cryptosystems Based On Discrete Logarithms Residues”. In Proceedings of the Advances in Cryptology– EUROCRYPT’99. LNCS, vol. 1592. Springer-Verlag.

Rubin A. D., (2002),”Security Considerations for Remote Electronic Voting”. Communications of the ACM, 45(12):39–44, December 2002.

Schoenmakers B., (1999),”A Simple Publicly Verifiable Secret Sharing Scheme and Its Application to Electronic Voting”. In Proceedings of the Advances in Cryptology – CRYPTO’99. Vol. 1666.

ABOUT THE AUTHORS

Okediran O. O. is a lecturer in the Department of Computer Science and Engineering, Ladoke Akintola University of Technology, Ogbomoso, Nigeria. He graduated with B.Tech. Computer Engineering and M. Tech. Computer Science from Ladoke Akintola University of Technology, Ogbomoso, Nigeria, in 2002 and 2008 respectively. He has almost finished his Ph.D Computer Science in the same Institution. He has published in reputable journals. His research interests include: Computational optimization, e-commerce, biometrics-based algorithms and their applications to e-voting systems. He belongs to the following professional bodies: Full member, Computer Professionals (Registration) Council of Nigeria (MCPN); Registered Engineer, Council for the Regulation of Engineering in Nigeria (COREN).

Omidiora E. O. is currently a lecturer in the Department of Computer Science and Engineering, Ladoke Akintola University of Technology, Ogbomoso, Nigeria. He graduated with B.Sc. Computer Engineering (1991) from Obafemi Awolowo University, Ile-Ife, Nigeria. He bagged M.Sc. Computer Science from University of Lagos, Nigeria (1998) and Ph.D Computer Science from Ladoke Akintola University of Technology (2006). He has published in reputable journals and learned conferences. His research interests include: The study of Biometric Systems, Computational Complexity measures and Soft Computing. He belongs to the following professional bodies: Full Member, Computer Professionals (Registration) Council of Nigeria; Corporate Member, Nigeria Society of Engineers; Register Engineer, COREN etc.

Olabiyisi S. O. received B. Tech., M. Tech and Ph.D degrees in Mathematics from Ladoke Akintola University of Technology, Ogbomoso, Nigeria, in 1999, 2002 and 2006 respectively. He also received M.Sc. degree in Computer Science from University of Ibadan, Ibadan, Nigeria in 2003. He is a lecturer in the Department of Computer Science and Engineering, Ladoke Akintola University of Technology, Ogbomoso, Nigeria. He has published in reputable journals and learned conferences. Dr Olabiyisi is a member of Computer Professional (Registration) Council of Nigeria (CPN). His research interests are in Computational Mathematics, Computational Complexity, Theoretical Computer Science, Simulation and Performance Evaluation.

Ganiyu R. A. is a lecturer in the Department of Computer Science and Engineering, Ladoke Akintola University of Technology, Ogbomoso, Nigeria. He graduated with B.Tech. Computer Engineering and M. Tech. Computer Science from Ladoke Akintola University of Technology, Ogbomoso, Nigeria, in 2002 and 2008 respectively. He has almost finished his Ph.D Computer Science in the same Institution. He has published in reputable journals. His research interests include: Dynamic Programming and their Applications; Theoretical Computer Science; Modelling and Simulation of Concurrent Systems Using Petri Nets (Low level and High level). He belongs to the following professional bodies: Full member, Computer Professionals (Registration) Council of Nigeria (MCPN); Registered Engineer, Council for the Regulation of Engineering in Nigeria (COREN).