UPGRADE is the European Journal for the Informatics Professional, published bimonthly at <http://www.upgrade-cepis.org/>

Publisher
UPGRADE is published on behalf of CEPIS (Council of European Professional Informatics Societies, <http://www.cepis.org/>) by Novática <http://www.ati.es/novatica/>, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática,* <http://www.ati.es/>)

UPGRADE monographs are also published in Spanish (full version printed; summary, abstracts and some articles online) by Novática

UPGRADE was created in October 2000 by CEPIS and was first published by Novática and INFORMATIK/INFORMATIQUE, bimonthly journal of SVI/FSI (Swiss Federation of Professional Informatics Societies, <http://www.svifsi.ch/>)

UPGRADE is the anchor point for UPENET (UPGRADE European NETwork), the network of CEPIS member societies' publications, that currently includes the following ones:
- InfoRewiew, magazine from the Serbian CEPIS society JISA
- Informatica, journal from the Slovenian CEPIS society SDI
- Informatik-Spektrum, journal published by Springer Verlag on behalf of the CEPIS societies GI, Germany, and SI, Switzerland
- ITNOW, magazine published by Oxford University Press on behalf of the British CEPIS society BCS
- Mondo Digitale, digital journal from the Italian CEPIS society AICA
- Novática, journal from the Spanish CEPIS society ATI
- OCG Journal, journal from the Austrian CEPIS society OCG
- Pliroforiki, journal from the Cyprus CEPIS society CCS
- Tölvumál, journal from the Icelandic CEPIS society ISIP

Cover page designed by Concha Arias-Pérez
"Indiscernible Identity" / © CEPIS 2010
Layout Design: François Louis Nicolet
Composition: Jorge Llácer-Gil de Ramales

Editorial correspondence: Llorenç Pagés-Casas <pages@ati.es>
Advertising correspondence: <novatica@ati.es>

UPGRADE Newslist available at
<http://www.upgrade-cepis.org/pages/editinfo.html#newslist>

The opinions expressed by the authors are their exclusive responsibility

ISSN 1684-5285

Monograph of next issue (April 2010)

"Information Technology inTourism Industry"

(The full schedule of **UP**GRADE is available at our website)

CEPIS UPGRADE

The European Journal for the Informatics Professional
http://www.upgrade-cepis.org

Vol. XI, issue No. 1, February 2010

Monograph: Identity and Privacy Management
(published jointly with Novática*)
Guest Editors: *Javier Lopez-Muñoz, Miguel Soriano-Ibañez, and Fabio Martinelli*

UPENET (UPGRADE European NETwork)

CEPIS NEWS

* This monograph will be also published in Spanish (full version printed; summary, abstracts, and some articles online) by **Novática**, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*) at <http://www.ati.es/novatica/>.

Privacy and Anonymity Management in Electronic Voting

Jordi Puiggalí-Allepuz and Sandra Guasch-Castelló

Security issues have to be considered when an electoral process is done electronically. Among them, one of the most important is voter privacy. Voter privacy is a requirement which is difficult to fulfil because this privacy requirement conflicts with other election accuracy requirements such as ensuring that all votes have been submitted by eligible voters. Methods that allow preserving election accuracy while preserving voter privacy are described in this article.

Keywords: Anonymity, Electronic Voting, Privacy.

Authors

Jordi Puiggalí-Allepuz is VP of Research and Development at Scytl Secure Electronic Voting, a multinational company with offices in Barcelona, Washington DC and Singapur, <http://www.scytl.com/>. He gained his MSc in Computer Science from the *Universitat Autònoma de Barcelona*, Spain, in 1994. He has participated in several European and Spanish funded research projects and has authored over 12 international publications and co-authored over 10 patents and patent applications. His fields of interest are cryptography and information privacy applied to electronic voting and e-government. <jordi.puiggali@scytl.com>

Sandra Guasch Castelló is a Researcher at Scytl Secure Electronic Voting. She gained her Bachelor Degree in Telecomunication specialized in Audiovisuals in 2006 and her Masters degree in Telecomunications in 2009 from the *Universitat Politècnica de Catalunya*, Spain. Currently she is working towards his PhD degree at the same center. She is co-author of 2 patents and two articles. Her fields of interest are advanced cryptography, privacy applied to electronic voting and peer-to-peer networks. <sandra.guasch@scytl.com>

1 Introduction

One of the most important security issues of an electoral process is to preserve voter privacy. In traditional electoral processes, this privacy is preserved using anonymous envelopes that are stored in a ballot box, and preventing them from being opened until the end of the electoral process. An Electoral Board ensures that the content of the ballot box is not manipulated to change the result of the election (i.e. destroying or adding votes).

The introduction of electronic voting increased the concerns related to voter privacy, such as the traceability of the voting transactions: it could be possible to correlate a voter with her vote if someone knew the time when this vote was submitted. Furthermore, there is not a physical control (i.e. visual) of the ballot box as in a traditional process. Therefore, people with privileged access to the voting system could be able to access the votes during the voting process and correlate them with voters. This privileged access could facilitate other malicious practices, like vote manipulation.

To mitigate these issues, the following security requirements are demanded in an electronic voting environment:

- **Vote authenticity:** it must be ensured that each vote is submitted by an eligible voter and that only one vote per voter is counted.
- **Voter privacy:** while it must be ensured that a voter is eligible, it must be impossible to correlate the voter identity with the content of a vote.
- **Accuracy of the election results:** it must be impossible for anyone to eliminate or modify the votes submitted by eligible voters or add invalid votes on behalf of other voters or non-eligible voters.
- **Privacy of the intermediate results:** the election intermediate results must be secret until the end of the process in order to prevent the influence of other voters who have not yet participated.
- **Vote verifiability:** the voters must be able to independently verify that their votes have been included in the final tally and have been correctly counted.
- **No coercion:** voters shall not be able to disclose their voting intent in order to prevent coercion or vote buying practices.

Therefore, it is necessary to ensure the authenticity of votes while the voters' privacy is preserved, two requirements which are difficult to solve at the same time.

The design of a system that fulfils these two requirements is difficult. However, this challenge could be solved using advanced cryptographic protocols like the ones presented in this article.

The article is structured as follows: in Section 2 basic cryptographic tools that are used in electronic voting and their limitations are explained, in Section 3 the different family types of cryptographic voting protocols and the way they manage the voters' privacy are presented. Finally, Section 4 contains the conclusions of this article.

2 Protection using Basic Cryptography

In a basic implementation, electronic voting systems use standard cryptographic techniques to ensure voter privacy and election accuracy. Usually, encryption and the digital signature of the vote are used in this case.

Vote encryption can be based on two types of cryptographic algorithms:

- Public key algorithms (or asymmetric): one election key pair is defined in these algorithms. One of the keys is public and known by all the voters, while the other is private and only known by an electoral authority (i.e. the Electoral Board). Voters use the public key to encrypt their votes, and when the voting process finishes, votes are decrypted using the private key in order to tally them. Therefore, nobody but the electoral authority can decrypt the encrypted votes.

- Secret key algorithms (or symmetric): in this type of algorithms the voter and the electoral entity share a unique secret key used for both operations: vote encryption and decryption. In order to prevent a voter from decrypting the votes of other voters, a unique secret key is generated for each voter. The electoral authority has access to all the keys.

Methods based on symmetric algorithms are not very scalable since the voting system should store a symmetric key for each voter. On the other hand, methods based on asymmetric algorithms have high computation costs for decrypting large messages. To solve these problems, hybrid systems are used. In these systems, the vote is encrypted using a random symmetric key (generated by the voter), that key is encrypted using the election public key. Therefore, the best characteristics of both types of encryption algorithms are combined. This technique is known as digital envelope.

The digital signature of a vote is applied over its encryption. Therefore, the verification of vote submission by an eligible voter is done using the voter's digital certificate. This technique is equivalent to the traditional election technique of signing an external postal envelope containing the vote: once the ballot is extracted from the envelope it is impossible to connect it with the signature. The ballot recovery from the postal envelope is equivalent to the vote decryption in an electronic voting environment. Therefore, decrypted vote contents cannot be linked to their signatures.

The security measures based on vote encryption and digital signature seem sufficient to protect voters' privacy. However, these measures are only efficient during the voting process. During the election tally, decrypted votes could be correlated with the voters who submitted them by checking the order in which votes are decrypted: the encrypted votes are digitally signed by the voters and therefore, decrypted votes can be correlated to the voter by checking the digital signature of the encrypted votes stored in their corresponding locations within the encrypted file.

In order to solve these problems, advanced cryptographic protocols have been designed for voting systems.

3 Protection using Advanced Cryptographic Protocols

There are some electronic voting advanced cryptographic protocols focused on the protection of voter privacy by means of vote anonymization. These protocols can be classified depending on the election phase where the anonymization techniques are implemented.

- **Pre-election phase:** these protocols implement the cryptographic processes to achieve anonymity during the election configuration process. They are basically focused on creating and assigning different credentials or anonymous paper ballots to each voter.
- **Voting phase:** these protocols implement the privacy protection processes during the vote casting step. The main objective of these protocols is to facilitate for voters the anonymous submission of the votes after these voters were properly identified.
- **Counting phase:** finally, these protocols are focused on achieving voter anonymity during the vote decryption and counting phase. These protocols prevent any correlation between the decrypted votes and their casting order.

Below, we will provide details of these systems.

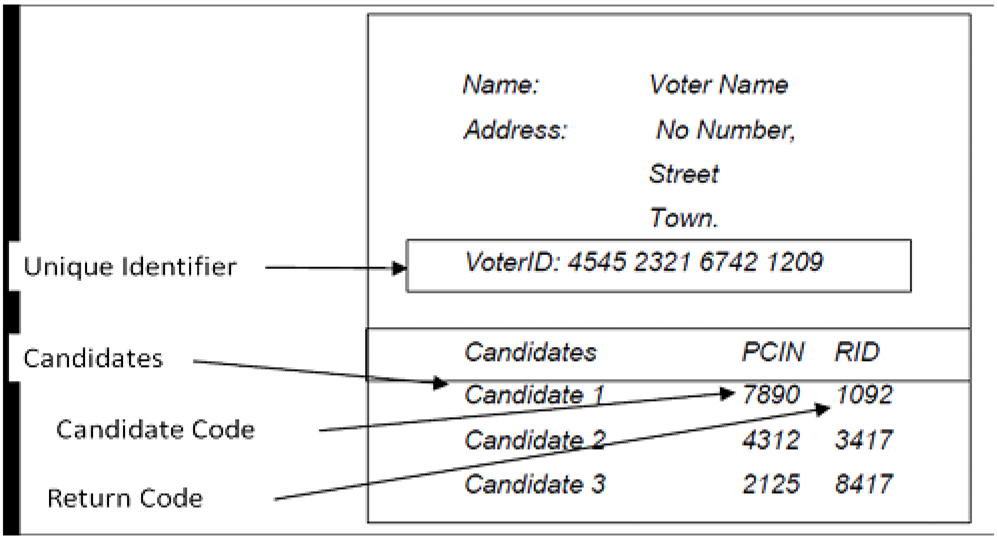

Figure 1: Pre-encrypted Paper Ballot.

3.1 Protocols which implement Voter Anonymity in the Pre-election Phase

In these protocols, the methods to preserve voter privacy are implemented in the phase of ballot design: a code is assigned to each voting option. This code is equivalent to the encryption of each option. The ballots with the pre-encrypted codes of the voting options are then sent to the voters.

These methods, generally known as *Pollsterless* [1], do not need any electronic voting device with cryptographic capacity to encrypt and cast a vote. Instead of this, the voter sends the pre-encrypted code of the selected voting option present in the received pre-encrypted ballot. An advantage of these methods, for example, is that they can be used to vote through mobile phones using SMS messages.

Assuming a traditional election, the mechanism used for designing the pre-encrypted ballots, is the following:

- A different code is assigned to each candidate or party. This code can be generated, for instance, using a Hash function applied on the voting option and a secret key related to a unique paper ballot identifier.
- Optionally, a second return code could be generated for each candidate or party. This code could be generated using the same Hash function but over the first code using a secret key only known by the voting server. This second code is used to verify the correct registration of the vote, as explained below.

Before the voting period starts the pre-encrypted ballots are randomly assigned to the voters and sent to them by postal service. Figure 1 shows an example of a paper ballot.

Once the voting process is open, the voter accesses the voting server and submits her vote by sending the paper ballot unique identifier (VoterID) and the codes belonging to the chosen candidates (PCIN). The voting server receives the vote and calculates the return codes (RID) using the received codes and its secret key. The voter receives the return codes (RID) and verifies that their value matches with those return codes assigned to her selected candidates in her pre-encrypted ballot. Since an attacker cannot know in advance the return codes assigned to the selected options, the voter can verify that her vote has been properly received by the server.

Finally, in the counting phase, the identity of the candidates related to the codes (PCIN) in the received votes is retrieved using the secret key assigned to the paper ballot identifier (VoterID).

From the point of view of privacy, these protocols allow:

- The submission of anonymous votes, since these are not digitally signed by the voters.
- The secrecy preservation of the voting options: the codes do not provide information about which candidate has been voted for, without having the pre-encrypted ballot.

However, this system is not perfect: there is still a chance of breaking the voter privacy if the pre-encrypted ballots are disclosed. For example, an attacker could know the vote intention of a voter using the submitted codes and the pre-encrypted ballot.

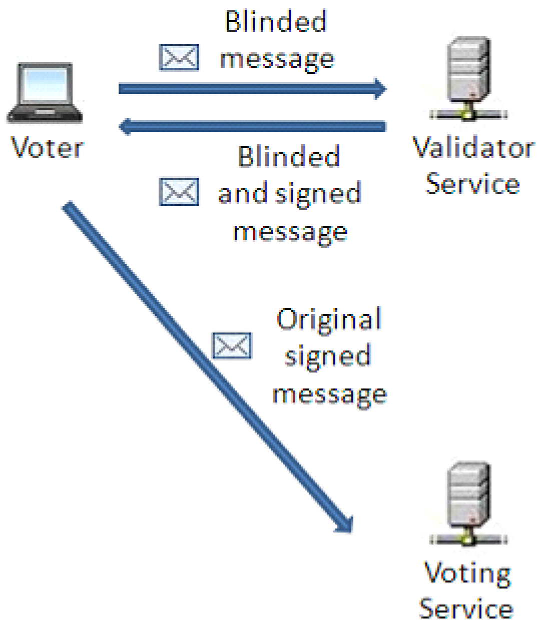

Figure 2: Two Agencies Model.

3.2 Protocols that Implement Anonymity in the Voting Phase

These protocols are focused on achieving an anonymous voting channel, allowing the voter to submit her vote without revealing her identity.

Generally, these protocols are based on what is known as the two agencies model [2], [3]. This model is based on the use of two independent services to identify the voter and submit her vote:

- The Validator Service: authenticates the voter, verifies her eligibility and allows her to vote in an anonymous way using an anonymous token.
- The Voting Service: receives encrypted votes with anonymous tokens from voters, and accept them after verifying if their tokens have been issued by the Validation Service.

It is usual to use the blind signature [4] property in these schemes, where the Validator digitally signs an encrypted vote without knowing exactly the contents it is signing. This property is based on the mathematic properties of some cryptographic algorithms such as RSA.

The voting process behaves as follows:

- The voter encrypts her vote using the election public key, "blinds" it and sends it to the Validator jointly with her voter credentials.
- The Validator receives the blinded message, verifies the eligibility of the voter, and digitally signs it, returning this signature to the voter.

- The voter receives the digital signature, removes the blinding factor and obtains the Validator digital signature of the encrypted vote.
- The voter sends then the encrypted and digitally signed vote to the Voting Service, which stores it after verifying if the digital signature belongs to the Validator Service.

In the Figure 2 a scheme of the system communication process is presented:

After the voting phase the votes are decrypted to perform the tally.

Therefore, these systems protect the voter privacy by means of:

- Obtaining a vote blindly signed by the Validator.
- Avoiding the voter identification at the vote submission time.
- Encrypting the voting options before sending them.

However, voter privacy still depends on the honesty of both agencies. Otherwise, they could collaborate, sharing information (i.e. IP directions) that would allow the correlation of the votes with the voters.

Also, there is a risk of election manipulation if the Validation Service is compromised, since it could forge encrypted and signed votes that would be successfully accepted by the Voting Service.

3.3 Protocols that implement Anonymity in the Counting Phase

These protocols encrypt their votes using an election public key and digitally sign the encrypted vote using voter digital certificates before casting them (see Section 2). However, in order to preserve voter privacy after vote decryption, these protocols also implement cryptographic mechanisms that prevent the correlation of the decrypted votes with the encrypted and digitally signed ones.

Depending on the cryptographic techniques used in the decryption process, these protocols can be divided into two types:

- Homomorphic tally protocols: these protocols obtain the final result of the election without decrypting the individual votes. Since the votes are never individually decrypted, their content cannot be correlated to the voters.
- Mixing protocols: these protocols break the correlation between the casting order of the encrypted votes and their decryption order, using decryption or re-encryption Mixing techniques.

3.3.1 Homomorphic Tally Protocols

These protocols use the homomorphic properties of some cryptosystems, by which certain operation over the encrypted votes is equivalent to the encryption of certain operation of the vote contents. In other words, if we have two votes, v_1 and v_2, and their encryptions $C(v_1)$ and $C(v_2)$, assuming that *Ö* and *È* are two algebraic operations, an homomorphic operation is defined as:

$$C(v_1)\ \ddot{O}\ C(v_2) = C(v_1\ \grave{E}\ v_2)$$

So, the result of the operation Ö of two encrypted votes $C(v_1)$ and $C(v_2)$ is equivalent to the encryption of the operation *È* of the votes v_1 and v_2. Depending on the type of operation, we have additive homomorphism (if is a sum) or multiplicative homomorphism (if is a product).

The additive homomorphism [5] is generally the most used since it generates the encrypted total sum of the votes, which is the desired result in a tally. Some algorithms have additive homomorphic properties, such as ElGamal (in the exponential version) or Paillier. In both algorithms, the product of the encrypted votes results in the encryption of the addition of the votes.

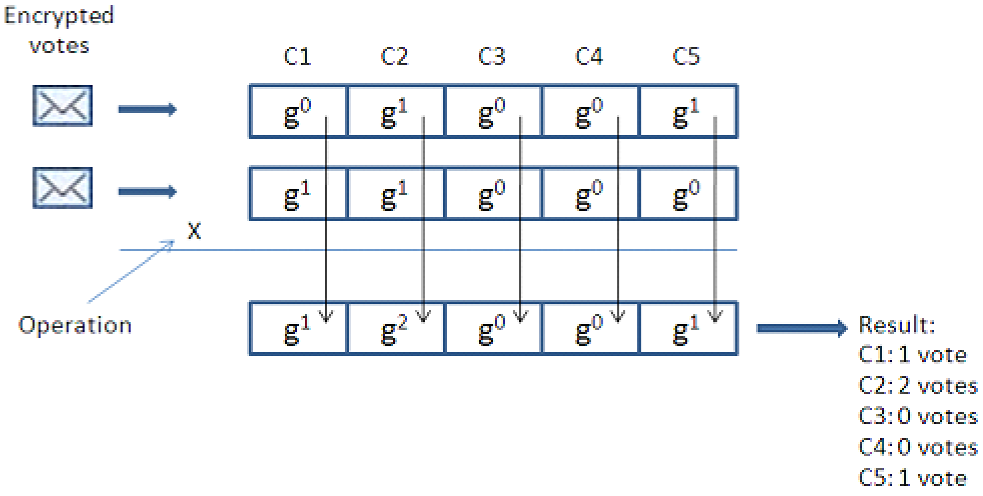

Figure 3: Additive Homomorphic Tally.

$$C(v_1) * C(v_2) = C(v_1 + v_2)$$

Therefore, just one decryption is required to obtain the sum of the votes. To achieve this property, votes must have a specific numeric format to obtain the sum of the votes for each candidate. This format is explained below.

In the multiplicative Homomorphism [6], the multiplication of the encrypted votes is equivalent to the encryption of the vote product.

$$C(v_1) * C(v_2) = C(v_1 * v_2)$$

These protocols are less used since they do not immediately give the total sum of votes after decrypting.

Depending on the desired type of Homomorphism, votes are represented in different ways. To explain a sample implementation, we will use as reference an additive homomorphic protocol based on ElGamal with exponentiation.

When additive Homomorphism is used, an encrypted vote is represented as a vector with as many elements as candidates in the election. In order to specify which candidates have or have not been selected by the voter, each vector element usually contains the value 1 or 0. In alternative implementations, the selections could be represented by the values 1 or -1. Then, the encrypted vote is calculated by individually encrypting the values of the vector elements. Since each element is individually encrypted, the encrypted values for each candidate can be operated separately. Thus, the number of times that a candidate has been selected can be obtained by decrypting the result of operating all the encrypted elements representing this specific candidate on all the cast votes. An example of this procedure is shown in Figure 3.

In exponential ElGamal, the vote is represented by the exponentiation of a public value *g*, common for all the voters, to the value 0 or 1. For example, assuming that a voter has selected the second and fifth candidates from five available, the vote could be represented as:

$$(g^0, g^1, g^0, g^0, g^1)$$

Since the exponents represent the candidate selection status in the votes, multiplying the votes we obtain the sum of the exponents. After decrypting each element of the vector, the number of selections of each candidate can be obtained by applying a logarithm in base *g* to each decrypted element.

Therefore, the protocols based on homomorphic tally provide a robust protection of the voter privacy, since they have the following characteristics:

- Votes are not individually decrypted, so there is no risk of correlating the content of individual votes with the vote casting order.
- Votes are encrypted by voters before submitting them.

The main problems of these systems are usually related to the preservation of election integrity. For example, since the votes are not individually decrypted, it is not possible to verify if a malicious voter has put more than one selection for a candidate (i.e. putting the exponent value 2 instead of 1 or 0). For this reason, these protocols require voters to prove that the encrypted vote has a valid selection (i.e., only 1 or 0 value) using zero knowledge cryptographic proofs. These proofs increase the computation cost in the voter terminal. If an encrypted vote is composed of as many encryptions as candidates, the computation cost of the proof proportionally increases according to the number of candidates. Therefore, these systems present scalability problems.

Finally, these protocols can only manage votes represented in a pre-fixed numeric format, so they cannot be used in elections where voters have to send selections or written answers.

3.2.2 Mixing Protocols

These protocols are based on reproducing the process in conventional elections where, at the end of the voting stage, the ballot boxes are shuffled to break the storage correlation order of the votes.

Once the correlation between voter and vote has been

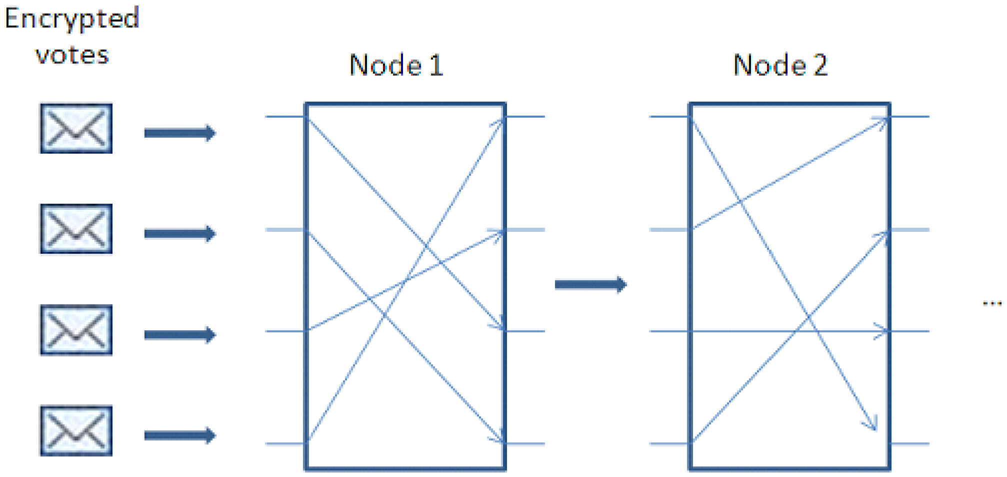

Figure 4: Example of a Net composed by Mix-nodes (Mixnet)

broken, votes can be decrypted in a secure way to obtain the results.

In these protocols, the shuffling process is based on a Mixing process. A Mixing process is a network of nodes (also called Mixnet), where each one permutes the votes received from a previous node and sends the permuted votes to the next node (see Figure 4). Since the permutation value is secret, the paths of the votes through the Mixnet cannot be guessed.

To prevent the disclosure of the paths of the votes through the Mixnet by comparing the input and output node values, the votes are also transformed at each Mix-node using re-encryption or decryption mechanisms. This changes the appearance of the encrypted votes without changing their original values [10]. Mixnets can be classified as:

- Decryption Mixnets: in these Mixnets the votes are nested encrypted by voters several times (as many times as nodes in the Mixnet), using the public key of each Mix-node in each encryption layer. When encrypted votes are provided to the Mixnet, each node permutes the input encrypted votes and uses its private key to remove one of the encryption layers (the one encrypted with its public key). This process is repeated at each node until it reaches the last one, where the last encryption layer is removed and the original vote contents are obtained. The main drawback of this method is that the voter has to encrypt her vote as many times as nodes in the Mixnet.
- Re-encryption Mixnets: this uses encryption algorithms that allow the re-randomization of a vote without adding another encryption layer. Therefore, although the re-encryption process of votes is executed many times, just one decryption step is required for obtaining the original votes. The advantage of this method is that the voter only needs to encrypt the vote once, since votes are re-encrypted at any node. The algorithms also have probabilistic properties (i.e. they add a random factor at each re-encryption, so two re-encryptions of the same vote are different), preventing any correlation of the inputs and outputs of the node after re-encryption and shuffling. Finally, a decryption step is done in the last node of the Mixnet to recover the plaintexts.

In these protocols the voter privacy is preserved by means of:

- the encryption of the votes by the voters.
- the use of a Mixing process that breaks the correlation between the signed and encrypted votes and the decrypted ones.

Therefore, voter privacy depends on the correct behaviour of the mixing process: if this does not permute properly the votes or manipulate them during the decryption/re-encryption process, the path of the votes through the Mixnet could be disclosed and the decrypted votes could be correlated with the voters.

There are some verification methods to verify the correct behaviour of the Mix-nodes [7]. Some of the most important are: Random Partial Checking [8] methods, systems that calculate secondary shuffles (alternative permutations and re-encryptions) or methods that use verification proofs of the correct permutation and re-encryption of the inputs at each Mix-node [9]. It is important that these verification processes are efficient and preserve the voter privacy. Since these processes are universally verifiable (i.e., any person without special privileges can verify that the process is correct), it is of paramount importance that they preserve voter privacy.

The main benefits of these protocols are that: they can use more flexible encryption schemes than homomorphic tally protocols; allow the use of hybrid encryption algorithms; support write-in; and have a better support of complex electoral processes.

4 Conclusions

Electronic voting systems introduce some challenging situations from the voter privacy point of view: it must be ensured that the votes belong to eligible voters while the voter intent must be kept as secret.

The use of standard cryptographic techniques like vote encryption and its digital signature solves this problem during the vote submission and storage steps. However, voter privacy is not fully guaranteed at the decryption phase, since the value of a decrypted vote could be connected to the encrypted and digitally signed vote.

This facilitated the introduction of advanced cryptographic protocols aimed at preserving voter privacy while ensuring election integrity (i.e. ensuring that the votes belong to eligible voters). These protocols can be classified depending on which phase of the election they are implementing the advanced cryptographic processes: during the election configuration, during the voting process or at the counting step. From these protocols, those that execute the anonymization process in the counting phase fit better with election security requirements. The main reason for this is that they provide a better control of election integrity (the encrypted votes are always digitally signed by the voters) and voter privacy processes (they can be executed in controlled and isolated environments). From these systems, Homomorphic tally protocols were initially the preferred choice from the voter privacy point of view, since they do not need to decrypt each vote individually for obtaining the election results. However, the evolution of the verification mechanisms in Mixing protocols and their flexibility in the management of complex elections, made them the preferred choice in complex elections with a large number of voters.

Finally, an interesting result of this study is that all the research in electronic voting is not restricted to this field. Therefore, it is expected that methods similar to mixing or homomorphic tally will be used in the future on other environments with high privacy demands.

References

[1] Tim Storer and Ishbel Duncan. "Two variations of the mCESG pollsterless e-voting scheme". The 29th Annual International Computer Software & Applications

Conference, 2005.

[2] D. Chaum. "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms". Communications of the ACM, vol. 24 no. 2, February, 1981.

[3] Atsushi Fujioka, Tatsuaki Okamoto, and Kazui Ohta. "A practical secret voting scheme for large scale elections". Advances in Cryptology - AUSCRYPT ’92, Lecture Notes in Computer Science, Berlin, 1993. Springer-Verlag.

[4] D. Chaum. "Blind Signatures for Untraceable Payments". D. Chaum, Advances in Cryptology Proceedings of Crypto 82, D. Chaum, R.L. Rivest.

[5] Josh Benaloh and Moti Yung. "Distributing the power of government to enhance the power of voters". In PODC. ACM, 1986.

[6] K. Peng. "A Hybrid E-Voting Scheme". 5th International Conference on Information Security Practice and Experience**,** 2009.

[7] Andreu Riera, Paul Brown, José Antonio Ortega. "Advanced Security to Enable Trustworthy Electronic Voting". 3rd European Conference on e-Government. 2003.

[8] Markus Jakobsson, Ari Juels, and Ronald L. Rivest. "Making mix nets robust for electronic voting by randomized partial checking". USENIX Security Symposium, 2002.

[9] C. Andrew Neff. "A verifiable secret shuffle and its application to e-voting". In ACM Conference on Computer and Communications Security. ACM, 2001.

[10] B. Adida. "Advances in Cryptographic Voting Systems". Thesis, 2006.

[11] T. W. Storer. "Practical Pollsterless Remote Electronic Voting". Thesis, 2007.