Secure Architectures for a Three-Stage Polling Place Electronic Voting System

Josué Figueroa González and Silvia B. González Brambila

Departamento de Sistemas, Universidad Autónoma Metropolitana,
Azcapotzalco, México DF,
Mexico

josue.figueroa@gmail.com, sgb@correo.azc.uam.mx

Abstract. Security on electronic voting systems is fundamental; it must assure the integrity of all the elements involved or generated during a voting process. This paper presents a design of secure architectures for providing security, integrity and authenticity of the most important elements involved in an electoral process: configuration files, recorded votes and final result files. Also, different cryptographic protocols for assuring security properties of configuration and final result files are presented as a part of one of the layers of the architectures. We consider a polling place electronic voting system composed by three stages and the use of three different systems during the whole process. Our analysis of architectures and protocols shows that the designed elements assure the secure properties which an electronic voting system must fulfill.

Keywords. Cryptographic protocol, electronic voting, integrity, secure architecture, security.

Arquitecturas de seguridad para un sistema de voto electrónico presencial de tres etapas

Resumen. La seguridad en los sistemas de voto electrónico es fundamental, esta debe asegurar la integridad de todos los elementos involucrados o generados durante el proceso de votación. Este trabajo muestra la creación de arquitecturas de seguridad para satisfacer la seguridad, integridad y autenticidad de los elementos más importantes involucrados en un proceso electoral: archivos de configuración, votos almacenados y archivo de resultados finales. Adicionalmente, como parte de una de las capas de las arquitecturas, se desarrollaron diferentes protocolos criptográficos que aseguran las propiedades de seguridad de los archivos de configuración y de resultados finales. Se considera un sistema de voto electrónico presencial formado por tres etapas y el uso de tres diferentes equipos durante todo el proceso. El análisis de las arquitecturas y de los protocolos muestra que los elementos diseñados aseguran las propiedades de seguridad que un sistema de voto electrónico debe satisfacer.

Palabras clave. Protocolo criptográfico, voto electrónico, integridad, arquitectura de seguridad, seguridad.

1 Introduction

This section presents a general description of an electronic voting system and a voting process for which the secure architectures presented in this paper were designed.

Electoral processes have been carried out using traditional methods like ballots or telephone calls, and many of the results are obtained through manual counting. As time passed and technology rose, first electronic voting systems appeared but they were considered only as electronic vote counters [9]. Nowadays, the use of these systems is quite common in many countries.

Electronic voting systems are divided into two groups: remote voting systems with voting performed in a location other than that of the voting center, and polling place voting systems with voting performed at the location point of voting equipment known as the electronic ballot box.

To complete the whole process, three systems have been created, each one fulfilling a special function:

– Configuration file generator that creates files containing the configuration of the elections to be carried out;

- Ballot box system that collects votes;
- Total result generator that collects all the results generated for each one of the ballot boxes and generates the final results for different elections.

The ballot box system issued in this paper is composed by three stages: the pre-voting stage, related to the installation of configuration files transported through a non-secure communication channel; the voting stage responsible for collecting votes, and finally, the post-voting stage that generates the final results for a particular equipment, which will be transported by a non-secure communication channel to a system that collects the final results of election.

Two kinds of users interact with the system: the functionary, a user responsible for the system (turns it on, configures it, and disables it) during the electoral process, and the voter, a principal user who casts his/her vote with the system.

1.1 Voting Process

A voting process begins with the creation of configuration files which contain information about elections. These files are produced by an agency responsible for conducting elections. These files are created before elections (the time depends on the regulations of the responsible agency).

The basic data of these files are: the name of elections to be performed and the options to choose from, which may include the names of political parties, candidates, questions of a poll, or others. Additionally, they can include information of the location of the ballot box, i.e., electoral district, state, etc. Validation of this information is made by the responsible agency and the persons involved in the election (candidates, representatives, volunteers, etc.). Digital signatures, public and private keys for the security of configuration files are generated in the same location where the configuration files were created.

Once configuration and security files have been generated, they are installed in the electronic ballot on the day of elections, during the electoral journey.

When the process of collecting votes is completed, a counting process begins, which can be of two types: total counting and partial counting. Partial counting gives the number of votes recorded in each one of the electronic voting machines; this is done at the place where the equipment is located. The results are displayed on a screen and also printed out so that the concerned audience can view them.

Total counting is performed in a place other than the location point of the ballot box, and with another system. The results of each partial counting are gathered and added to obtain the final results for each election. The results of each ballot box can be sent to this system through Internet or delivered using a storage device.

2 Related Work

This section discusses work related to the security of electronic voting systems.

When an electronic voting system is constructed, cryptography is not a problem; there exist many cryptographic techniques which have been efficiently tested. The problem appears when a system is developed under a non-secure platform or architecture; such problem is known as “the secure platform problem” [8]. For an electronic voting system, the data to be protected are votes considered as the fundamental element of the system, and also the so-called critical data which include configuration files and final result files [5]. According to [10], the menaces to consider at the moment of developing an electronic voting system are the following:

- External attacks. Until this moment, external attackers have not had enough time to access the systems for altering them; this is explained mainly by a lack of external ways to access the systems;
- Malicious voters. A voter might try to obtain an improper access to the system and vote more than once or affect the system performance.

There are a lot of papers about security on electronic voting systems including such topics as security protocol development [6], secure architectures [2, 10, 11] and the right way for performing an electoral process [4, 12]. Also,

there are papers that analyze existing voting systems [5].

A protocol similar to the one presented in this paper can be found in [6]; however, the latter work is focused more on remote voting. It manages the security of votes during a voting process efficiently, but it does not make any reference to the security of critical data (after or before a voting process), which are an important part of the system.

The same is true of the approaches presented in [2, 10, 11]. They are focused on remote voting and management of security for votes during their transmission to another system. The referenced papers do not consider the security of the pre- and post-voting stages.

Polling-place voting is explored in [1]; here, the security of the votes is linked to the activation method. Also, the security of data after the election ends is assured by a cryptographic protocol; however, this approach does not consider the security of configuration files.

3 Development

In this paper, the following attacks to be solved are considered:

- Modification of configuration files, which can alter the way votes are recorded;
- Relationship between the vote and the voter, so that someone could know if a particular voter votes for a particular candidate or option;
- Possibility for a voter to cast more than one vote;
- Modification of final result files, which alters the results of election.
- We do not consider such attacks as:
- Supplanting a voter (this attack can be realized on a system that activates the ballot box) or cheating a functionary;
- Changing a vote. This attack does not affect votes inside the system, which are secure because an attacker has no way to access them. However, when the results are outside the system, they can be threatened.

Table 1 presents the nomenclature used in the protocols. Note that when a *-1* is used for any key in the protocols means that it is used for decrypting information.

After reviewing the architecture in [7], it was found that its layers and operation were appropriate for an electronic voting system. The cryptography layer is composed of cryptographic primitives and protocols that assure secure properties of an electronic voting system [2].

Table 1. Nomenclature for different elements used in the protocols

Equipment	Key	Nomenclature
Generation of configuration files	Encrypted public key	$\hat{}e_{GM}$
	Private key	d_{GM}
	Symmetric key	k_{GM}
	Special key	k_{ESP}
Ballot box	Public key	e_U
	Private key	d_U
	Symmetric key	k_U
	Private key for signing	d_{Uf}
	Public key for verifying	e_{Uf}
Obtaining total results	Public key	e_R
	Private key	d_R
	Symmetric key	k_R
	Special key	k_{ESP}
	Captured result key	r_T

3.1 Initial Considerations

For developing secure architectures and protocols, an electronic voting system of three stages mentioned in Introduction was considered. For a correct implementation of protocols, it is necessary to install some keys on different equipment before the electoral process begins. This initial distribution is shown in Table 2. Symmetric and asymmetric keys not shown in Table 2 are generated when necessary. Such generation does not impact the system

performance. Public and private keys take longer time to be generated, but there are only few of them. There are much more symmetric keys — about 500-700 times — that are generated and stored really fast. Tests on a PC take less than 2 seconds, and in an embedded system, less than 5 seconds. The fact that symmetric keys are generated when they are needed increases the security of stored votes.

Table 2. Initial location for different keys involved in the protocol

Generation of configuration files	Ballot box	Equipment that obtains total results
(e_U) (d_{GM})	(d_U) (^e_{GM}) (e_R)	(d_R) (e_R)

There is no limit for the amount of equipments to be installed; also, there is no restriction which ties a physical device with the configuration information that can be installed on it.

3.2 Pre-Voting Stage Security

At this stage, security must assure the integrity and authenticity of received configuration files, that is, these files must not have been modified or substituted. Security of generated records during system configuration has to be guaranteed also. Secure architecture for this stage is composed of two layers: the control access layer and the cryptography layer.

The control access layer allows or denies access to the configuration interface and sends the generated elements to the cryptography layer. Components of the control access layer are the following:

- Subject: a functionary;
- Secure object: configuration interface;
- Authorization: allows or denies access to the configuration interface;
- Restrictions: the turning-on date and hour must be posterior (within certain limits) to the ones registered in the system for beginning its operation.

Using a cryptographic protocol, the cryptography layer validates the integrity and authenticity of configuration files.

It can detect if these files have been modified or if they proceed from a different source than the authorized one. Also, using symmetric ciphering, it assures the security of generated elements.

Cryptographic Protocol

The cryptographic protocol is divided into two main steps: generation and verification.

Generation Protocol

This step is performed by the configuration file generator, and the protocol must guarantee that any alteration of files will be detected. Also, if an attacker creates a new set of files, these must be detected as non-valid.

Generating a set of digital signatures (s_1, s_2, s_3, , s_n) for each one of the configuration files (1), a special key (k_{ESP}) is created by taking parts of these signatures (2). Using the special key, the public key of the configuration file generator (e_{GM}) is encrypted (3). The configuration files are ciphered (a) using the symmetric key of the configuration file generator (k_{GM}) (4); this key is protected by its ciphering with the ballot box public key (e_U) (5). Once finished, the files which will be sent are: the encrypted data (c), the digital signatures (s) and the encrypted symmetric key (p). Here are the steps of the protocol:

1: $s = d_{GM}(a, H(a))$.
2: $k_{ESP} = s_1 + s_2 + s_3 + \ldots + s_n$
3: $^{e_{GM}} = k_{ESP}(e_{GM})$
4: $c = k_{GM}(a)$.
5: $p = e_U(k_{GM})$.

Verification Protocol

When the set of files {*c, s, p*} is received, the encrypted symmetric key (*p)* is decrypted with the ballot box private key (d_U) so that the symmetric key (k_{GM}) is obtained (1); the latter deciphers the encrypted data (c) so that the configuration files (a) are obtained (2). Using the set of digital signatures (s_1, s_2, s_3, … , s_n), the special key (k_{ESP}) is created (3). This key deciphers the public key

of the configuration file generator (e_{GM}) (4). If the digital signatures have not been altered, the special key (k_{ESP}) will correctly decipher the public key (e_{GM}), and the integrity of data is thus assured (5). Here are the steps of the protocol for verifying the integrity and authenticity of data:

1: $k_{GM} = d_U$ (p).
2: $a = k_{GM}^{-1}$ (c).
3: $k_{ESP} = s_1 + s_2 + s_3 + \ldots + s_n$
4: $e_{GM} = k_{ESP}^{-1}(\hat{}e_{GM})$
5: e_{GM} (H(a),s).

3.3 Voting Stage Security

A secure architecture for this stage must guarantee the integrity and confidentiality of stored votes. Such architecture is composed of three layers: the authentication layer, the control access layer, and the cryptography layer.

The authentication layer validates the identity of a user determining if the user is allowed to participate according to the following restrictions: the user must be registered in the list containing voters allowed to vote. Also, the user must not have participated previously. These conditions are verified on another system not considered in this paper.

The control access layer is based on information provided by the authentication layer; it determines the type of the user who will interact with the system and at this stage is expected to be a voter. Once the voter has completed his/her participation, the system is disabled so that this voter cannot vote again. The elements of the control access layer are the following:

- Subject: a voter;
- Secure object: a voting interface and a file with registered votes;
- Authorization: allows the access to the voting interface according to some restrictions;
- Restrictions: a voter must be validated for participation and can participate in voting only once; the system can record votes only until a specified hour.

There are two possibilities to prevent double participation of a voter.

The first possibility involves the activation step, when the voter must be validated before voting by a functionary in a separate system which possesses the information of the voters.

The second possibility involves the final step, when the voter finished his participation.

The fact of the voter's participation is registered by the system, and if he wants to participate again, the system will reveal this intent, and the functionary will not enable the ballot box. Thus, when the voter is permitted to vote, this enables the system to register the vote. After that, the system is disabled and does not allow the voter to access the interface.

The cryptography layer uses symmetric ciphering algorithms to assure the integrity of registered votes. A different key, of an appropriate length [3], is used for ciphering each vote. The votes are stored using random storage in order to avoid the relationship 'vote – voter'.

The keys used for ciphering votes are generated when the system is configured during the pre-voting stage. When a vote is registered, the symmetric key is chosen randomly, and the vote is encrypted before being stored.

3.4 Post-Voting Stage Security

At this stage, a secure architecture assures that once the system has been turned off, it cannot be turned on for introducing more votes. It also manages the security, authenticity and integrity of the files that will be sent to another system and of the file of generated records. Such architecture is formed by three layers: the authentication layer, the control access layer, and the cryptography layer.

The authentication layer validates the identity of a user determining if the user is a functionary, and allows such user to access the administration interface in order to finalize the electoral process.

The control access layer is based on the information sent by the authentication layer and determines if a particular user is a functionary. When the electoral process ends, this layer disables the system so that it cannot be used again. The elements of this layer are the following:

- Subject: a functionary;
- Secure object: a management interface and a file with registered votes;

- Authorization: allows the access to the administration interface according to some restrictions;
- Restrictions: only a functionary can access the management interface, and optionally, it can be accessed only after a specified hour.

The cryptography layer deciphers stored votes for final counting and obtaining the results of each election. This layer applies a cryptographic protocol for assuring the integrity and authenticity of the result file which will be sent to the final result generating system.

Cryptographic Protocol

At this stage, the encrypted items are those generated during the whole process, i.e., votes, final result, and records which indicate that the electoral process has ended.

The cryptographic protocol is divided into the same steps as the pre-voting stage protocol.

Generating Protocol

This protocol must assure that received data are the same that the ballot box generated. A set of asymmetric keys is created in the ballot box *(d_{Uf}* and *e_{Uf})* (1) for generating a digital signature *(s)* of the results file *(r)* (2). This file and the private key *(d_{Uf})* are encrypted using the symmetric key *(k_U)* (3) obtaining *(c)* and *(^d_{Uf})* (4). After that, the symmetric key *(k_U)* is encrypted using the public key *(e_R)* producing *(p)* (5). A special key *(K_{ESP})* is formed with the digital signature (6) which is used for ciphering the public key *(e_{Uf})* obtaining *(^e_{Uf})* (7). After these steps, the set of files to be sent are: *{c, ^e_{Uf}, ^d_{Uf}, p}*. The steps of the protocol are the following:

1: d_{Uf}, e_{Uf}
2: s = d_{Uf} (r, H (r)).
3: c = k_U (r).
4: ^d_{Uf} = k_U (d_{Uf}).
5: p = e_R (k_U).
6: k_{ESP} = s(r).
7: ^e_{Uf} = k_{ESP}(e_{Uf}).

Verifying Protocol

Once the set of files *{c, ^e_{Uf}, ^d_{Uf}, p}* is received, the symmetric key *(k_U)* is decrypted using the final result generating system private key *(d_R)* (1).

Then with *(k_U)*, the encrypted results *(c)* and result digital signature are decrypted (2). Also using this key, the private key (d_{Uf}) is decrypted (3). After that, results are captured from the record *(r_T)* and their digital signature *(s)* is obtained using *(d_{Uf})* (4). This key is used for creating *(k_{ESP})* (5) which deciphers the public key *(e_{Uf})* (6) which in its turn verifies the integrity of the results (7). The steps for assuring authenticity and integrity are the following:

1: k_U = d_R (p).
2: r = k_U-1 (c).
3: d_{Uf} = k_U-1 (^d_{Uf}).
4: s = d_{Uf}(r_T).
5: k_{ESP} = s.
6: e_{Uf} = k_{ESP}(^e_{Uf}).
7: e_{Uf} (H(r), s).

4 Security Tests

In order to test the performance of security architectures and cryptographic protocols, an electronic ballot box was developed, in which the architecture's elements and protocols were implemented.

4.1 Pre-Voting Stage Security

The control access layer did not allow the system to be used before the indicated hour and showed a message indicating that the system was turned on at a wrong moment and turned it off automatically.

For testing efficiency of the cryptographic protocol, a new set of configuration data, public and private keys were created. All possible cases of combining this data were tested, even assuming that an attacker obtained the original set of data. The goal was that the system recognized the altered data as valid. The results for each case are presented below, cases 1-6.

As it can be seen, the only case in which the system recognized data was when the entire original set of data files was used. Any other combination produced an error which was detected by the system.

Case 1	
Original data: --- Modified data: d_{GM}, e_U, s, a	
Verifying	
$k_{GM} \neq d_U(p)$	ERROR: The attackers' public key doesn't match with the original private key.

Case 2	
Original data: e_U Modified data: d_{GM}, s, a	
Verifying	
$k_{GM} = du(p)$ $a = k_{GM}^{-1}(c)$ $e_{GM} \neq K_{ESP}^{-1}(\hat{}e_{GM})$	The public key matches with the private key. Data are correctly decrypted. ERROR: The public key (e_{GM}) is not deciphered correctly because the signature creates a different special key than the expected one.

Case 3	
Original data: e_U, s Modified data: d_{GM}, a	
Verifying	
$k_{GM} = du(p)$ $a = k_{GM}^{-1}(c)$ $e_{GM} = K_{ESP}^{-1}(\hat{}e_{GM})$ $e_{GM}(a,s)$	The public key matches with the private key. Data are correctly decrypted. The public key is correctly decrypted. ERROR: Modified data do not match with the original signature.

Case 4	
Original data: e_U, d_{GM} Modified data: s,a	
Verifying	
$k_{GM} = du(p)$ $a = k_{GM}^{-1}(c)$ $e_{GM} \neq K_{ESP}^{-1}(\hat{}e_{GM})$	The public key matches with the private key. Data are correctly decrypted. ERROR: The public key (e_{GM}) is not correctly decrypted because the modified signature does not create the right special key.

Case 5	
Original data: e_U, d_{GM}, s Modified data: a	
Verifying	
$k_{GM} = du(p)$ $a = k_{GM}^{-1}(c)$ $e_{GM} = K_{ESP}^{-1}(\hat{}e_{GM})$ $e_{GM}(a,s)$	The public key matches with the private key. Data are correctly decrypted. The public key is correctly decrypted. ERROR: Verification fails because data do not match with the original signature.

Case 6	
Original data: e_U, d_{GM}, s, a Modified data: ---	
Verifying	
$k_{GM} = du(p)$ $a = k_{GM}^{-1}(c)$ $e_{GM} = K_{ESP}^{-1}(\hat{}e_{GM})$ $e_{GM}(a,s)$	The public key matches with the private key. Data are correctly decrypted. The public key is correctly decrypted. SUCCESS: Data match with the original signature.

4.2 Voting Stage Security

The authentication layer never allowed a voter to participate more than once, and the control access layer disabled the system each time when a voter finished his/her participation to prevent double participation.

The cryptography layer encrypted each vote with its own key thus raising the security levels of the system, and random storage did not allow the relationship between a voter and his/her vote.

4.3 Post-Voting State Security

The authentication layer validated a functionary correctly in all cases, and the control access layer never allowed a voter to access the management interface. The elements of the cryptographic layer maintained the security of generated elements. The protocol for assuring integrity and authenticity

of the result file was tested with the same kind of tests used at the pre-voting stage.

The protocol was subjected to the following tests with the results presented in cases 7-11.

Case 7	
Original data: --- Modified data: s, r, d_{UF}, e_{UF}, e_R	
Verifying	
$k_U \neq d_R$ (p)	ERROR: The public key does not match with the private key.

Case 8	
Original data: e_{Uf},e_R Modified data: s, r, d_{Uf}	
Verifying	
$k_U = d_R(p)$ $r=k_U^{-1}(c)$ $d_{Uf} = k_U^{-1}(\hat{}d_{Uf})$ $s=d_{Uf}(r_T)$ $k_{ESP} = s$ $e_{Uf} \neq k_{ESP}(\hat{}e_{Uf})$	The public key matches with the private key. Results are correctly decrypted. The private key is correctly decrypted. Results from the final record are captured and signed. The special key is created. ERROR: The special key is not the expected one because data have been modified and the public key cannot be decrypted.

Case 9	
Original data: e_{Uf}, e_R, d_{Uf}, s Modified data: r	
Verifying	
$k_U = d_R(p)$ $r = k_U^{-1}(c)$ $d_{Uf} = k_U^{-1}(\hat{}d_{Uf})$ $s = d_{Uf}(r_T)$ $k_{ESP} = s$ $e_{Uf} = k_{ESP}(\hat{}e_{Uf})$ $e_{Uf}(r,s)$	The public key matches with the private key. Results are correctly decrypted. The private key is correctly decrypted. Results from the final record are captured and signed. The special key is created. The special key is the expected one and deciphers the public key. ERROR: The special key is not the expected one because data have been modified and the public key cannot be decrypted.

Case 10	
Original data: e_{Uf}, e_R, d_{Uf} Modified data: s, r	
Verifying	
$k_U = d_R(p)$ $r = k_U^{-1}(c)$ $d_{Uf} = k_U^{-1}(\hat{}d_{Uf})$ $s=d_{Uf}(r_T)$ $k_{ESP} = s$ $e_{Uf} \neq k_{ESP}(\hat{}e_{Uf})$	The public key matches with the private key. Results are correctly decrypted. The private key is correctly decrypted. Results from the final record are captured and signed. The special key is created. ERROR: The special key is not the expected one because data have been modified and the public key cannot be decrypted.

Case 11	
Original data: e_{Uf}, e_R, d_{Uf}, s, r Modified data:	
Verifying	
$k_U = d_R(p)$ $r = k_U^{-1}(c)$ $d_{Uf} = k_U^{-1}(\hat{}d_{Uf})$ $s=d_{Uf}(r_T)$ $k_{ESP} = s$ $e_{Uf} = k_{ESP}(\hat{}e_{Uf})$ $e_{Uf}(r,s)$	The public key matches with the private key. Results are correctly decrypted. The private key is correctly decrypted. Results from the final record are captured and signed. The special key is created. The special key is the expected one and deciphers the public key. SUCCESS: Verification is valid because all data are original and have not been modified.

The obtained results show that only when the original set of signatures, keys, and results are used, the system recognizes the data as valid.

5 Conclusions and Future Work

The electronic voting process involves more than collecting and counting of votes but also management of files involved during the whole process which is important too. Security of all elements used and generated during the configuring stage, vote gathering and final counting is fundamental for these types of systems. The secure architectures designed for each stage of the voting process assure the most important secure properties to be fulfilled by any electronic voting system. They guarantee that a

voter cannot vote more than once and that a vote cannot be related with the voter who issued it. The tests of the protocols were focused more on the steps that conform these protocols than on the security of the algorithms that are used at each step. It is important to take into account that polling place systems are considered secure because it is difficult for an attacker to obtain control of them; however, they are vulnerable at the moment of sending the configuring information or when results are sent to another system. As it is shown in the section devoted to the tests, the designed protocols are able to detect any modification of this critical data, even when an attacker gains access to them or to different keys used during the whole process.

The main difference between our research and other works related to electronic voting security, besides the fact that the latter are more focused on remote systems, is that most of these papers deal only with vote security, but few of them consider the so-called critical files. For an attacker, it can be difficult to access votes during the system operation; however, accessing the files while these are outside the system may be easier. An attack on configuration or result files can alter results without even altering votes. This paper presents a method to prevent such attacks, especially considering the fact that configuration or result files can be transported through an insecure communication channel. The security management of all elements generated throughout the process of voting is the main contribution of tour work.

Future work within this approach may include development of secure architectures for the stage of configuration file generation and the stage of obtaining total results, and studying the way these are related to the architectures presented in this paper.

References

1. **Clausen, D., Puryear, D., & Rodriguez, A. (2000).** *Secure voting using disconnected distributed polling devices*. Palo Alto. CA: Standford University.
2. **Cranor, L.F. & Cytron, R.K. (1997).** A security-conscious electronic polling system for the Internet. *Thirtieth Hawaii International Conference on System Sciences*. Wailea, HI, USA, 3, 561–570.
3. **García, L., Morales, G., & González, S.B. (2005).** Implementación del algoritmo RSA para su uso en el voto electrónico. *Simposio acerca de las urnas electrónicas para la emisión del voto ciudadano*, México, D.F, 59–69.
4. **Kohno, T., Stubblefield, A., Rubin, A.D., & Wallach, D.S. (2004).** Analysis of an electronic voting system. 2004 *IEEE Symposium on Security and privacy*. Berkeley, CA., USA, 27–40.
5. **Liaw, H.T. (2004).** A secure electronic voting protocol for general elections, *Computers & Security*. 23(2), 107–119.
6. **Probst, S., Essmayr, W., & Weippl, E. (2002).** Reusable Components for Developing Security-Aware Applications. *18th Annual Computer Security Applications Conference*, Las Vegas, Nevada, USA, 239–248.
7. **Rivest, R.L.** Electronic Voting, Retrieved from http://theory.lcs.mit.edu/~rivest/Rivest-ElectronicVoting.pdf.
8. **Saltman, R.G. (2003).** Auditability of non-ballot, poll-site voting systems. Retrieved from http://vote.nist.gov/pospapers/Saltman-AuditabilityofDREs%28Revised%292003.pdf.
9. **Selker, T. & Goler, J. (2004).** The SAVE System: Secure Architecture for Voting Electronically, *BT Technology Journal*, 22(4), 89–95.
10. **Thomas, N. (2005).** Performability of a secure electronic voting algorithm. *Electronic Notes in Theoretical Computer Science*, 128(4), 45–58.
11. **Voting Technology Project (2004).** *Insuring the integrity of the electoral process: recommendations for consistent and complete reporting of election data*. Pasadena, CA, Caltech/MIT.
12. **Williams, B.J. & King, M.S. (2004).** Implementing voting systems: the Georgia method, *Communications of the ACM*, 47(10), 39–42.

Josué Figueroa González is an electronic engineer by the Metropolitan Autonomous University (UAM), campus Azcapotzalco, Mexico, where he also received a M. S. degree in Computer Science. He worked as a professor and researcher at the Systems Department of the same university from 2007 to 2011. He is certified in Microsoft Project and Java Associated. His research interests include security and confidence on electronic voting and IT Governance.

Silvia Beatriz González Brambila holds a Computer Science degree from the Metropolitan Autonomous University (UAM), campus Iztapalapa, Mexico, a M.S. degree in Computer Science from UAM, campus Azcapotzalco, Mexico, and a Ph.D. in Computer Science from the Technological Research Institute of Monterrey, Morelos, Mexico. She works as a professor and researcher at the Systems Department of UAM, campus Azcapotzalco. She was the coordinator of the Computer Science Master Degree Program, the Computer Engineering Program, a teaching coordinator at the Basic Sciences and Engineering Department of UAM, campus Azcapotzalco, and EGEL of Informatics and Computer. Her research interests include electronic voting, IT governance and visual learning for robots.

Article received on 17/11/2010; accepted on 29/09/2011.