Yet Another Improvement over the Mu-Varadharajan e-voting Protocol

Ortiz-Arroyo, Daniel; Rodriguez-Henriquez, Francisco

Published in:
Computer Standards & Interfaces

Publication date:
2007

Document Version
Early version, also known as pre-print

Available online at www.sciencedirect.com
ScienceDirect
Computer Standards & Interfaces 29 (2007) 471–480
COMPUTER STANDARDS & INTERFACES
www.elsevier.com/locate/csi

Yet another improvement over the Mu–Varadharajan e-voting protocol

F. Rodríguez-Henríquez [a,*,1], Daniel Ortiz-Arroyo [b], Claudia García-Zamora [a,1]

[a] *Computer Science Department, Centro de Investigación y de Estudios Avanzados del IPN Av. Instituro Politécnico Nacional No. 2508, México D.F., México*
[b] *Computer Science and Engineering Department Aalborg University Esbjerg Niels Bohrs Vej 8, 6700 Esbjerg Denmark*

Received 10 January 2006; received in revised form 3 October 2006; accepted 11 November 2006
Available online 9 January 2007

Abstract

In this paper we present a fully functional RSA/DSA-based e-voting protocol for online elections that corrects and improves a scheme previously proposed by Lin- Hwang-Chang [I. Lin, M. Hwang, C. Chang, Security enhancement for anonymous secure e-voting over a network, Comput. Stand. Interfaces 25 (2) (2003) 131–139.]. We found that Lin-Hwang-Chang's scheme and a recent modification of it by Hwang-Wen-Hwang [S. Hwang, H. Wen, T. Hwang, On the security enhancement for anonymous secure e-voting over computer network, Comput. Stand. Interfaces 27 (2) (2005) 163–168.] have an important weakness. Moreover, the scheme proposed by Yang-Lin-Yang [C. Yang, C. Lin, H. Yang, Improved anonymous secure e-voting over a network, Information and Security 15 (2) (2004) 185–191.] also suffers from this same problem. We describe in detail our findings and propose a new scheme to overcome the weakness we found in these schemes effectively. Finally, we describe the implementation details of our protocol and present its preliminary performance evaluation.

Keywords: Electronic voting; Cryptography; Blind signature; DSA; RSA

Contents

* Corresponding author. Tel.: +52 55 50613758; fax: +52 55 50613757.
E-mail addresses: francisco@cs.cinvestav.mx (F. Rodríguez-Henríquez), do@cs.aaue.dk (D. Ortiz-Arroyo).
[1] Work partially supported by CONACYT project 45306, México.

0920-5489/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.csi.2006.11.003

1. Introduction

Democratic societies, parliaments, company boards, syndicates and other similar organizations need to provide convenient and secure mechanisms for voter members to cast their ballots during elections. In elections, voters may express freely their political preferences. Similarly, in referendums and opinion polls the public opinion is sensed on issues of general interest. In all these democratic processes it is essential that the voting system provides privacy, security, and accuracy during vote counting, to guarantee fairness and secrecy. It is also important to facilitate easy access for voters to the election polls. However, reaching all these objectives may be very difficult to achieve because of economic and administrative constraints.

Recent advances in communication networks and cryptographic techniques have made possible to consider on-line voting as a feasible alternative to conventional elections. On-line electronic voting has the flexibility of allowing users to participate in an election no matter where they physically are at the moment of the voting process. The only requirement for a voter to participate in an election is to have a means of establishing a wired or wireless Internet connection to the servers of the voting system. Additionally, an aggregated value of this kind of system is the inherent privacy they provide, since a user can participate actively within an election process without being seen by anyone. Achieving this level of privacy would be almost impossible in a traditional election system.

Creating a secure on-line voting system requires the use of robust security mechanisms that are relatively complex to design. Accordingly, the study of security schemes in electronic elections has received considerable attention in the last twenty years. As a result of this interest, a wide variety of e-voting cryptographic protocols have been proposed [4–8,1,9,10]. Such protocols must satisfy a number of desirable security requirements such as: vote accuracy, verifiability, voters' privacy, and double voting detection among others [1,9]. Roughly speaking, the cryptographic protocols that have been proposed can be classified as the ones based on *homomorphic functions*, and the ones based on *blind signatures*.

The design of protocols based on *homomorphic functions*, requires rather complicated encryption schemes for hiding ballot's content in order to preserve voters' privacy [11,10]. Those protocols include two phases: ciphering and voting. Several techniques such as shared secret keys and zero-knowledge proofs have been proposed in order to implement those two phases.

Blind signatures were proposed in 1983 by Chaum [4]. Protocols based on blind signatures hide voter's identity, but still make the actual content of a vote visible to the authority. Protocols based on blind signatures generally consist of a registration phase followed by a voting phase. An example of a protocol based on blind signatures is Mu and Varadharajan [12] e-voting scheme.

The main contribution presented in this paper is to show that all variations of the Mu and Varadharajan [12] e-voting scheme that have been proposed so far, namely, the Lin et al. [1], Hwang et al. [2] and Yang et al. [3] schemes, share a functional flaw: a high possibility of not being able to sign the voting content m. Moreover, we show that this difficulty arises from the way that all those schemes use the ElGamal digital signature algorithm. Furthermore, we propose a new modification to these schemes which allows overcoming the security flaw we found effectively.

Our solution employs the Digital Signature Algorithm [13] together with the RSA scheme for generating several blind signatures, rather than the ElGamal signature scheme used in previous e-voting schemes. We illustrate our solution by describing in detail how to modify the scheme proposed by Lin et al. [1]. Our solution consisting of replacing the ElGamal signature algorithm with the DSA algorithm can be easily extended to the schemes in [2,3] too. However, the specific details are not covered in this contribution.

The rest of this paper is organized as follows. Section 2 briefly describes related work on e-voting schemes. Section 3 reviews one of the three Mu and Varadharajan scheme variations known: the Lin et al. modification. Although the material in Section 3 is mainly focused on the description of the Lin et al. scheme, we briefly explain the variations proposed by Hwang et al. and Yang et al. as well. Additionally, in that section, the functional flaw present in those three schemes is explained. Based on the observation of that weakness, in Section 4 we propose an improved RSA/DSA-based of Lin et al. security protocol for online voting. Next Section describes implementation details of the system developed. A brief performance evaluation of the system is presented in Section 6. Finally, in Section 7 some concluding remarks are drawn.

2. Related work

Blind signatures were proposed in 1983 by Chaum [4]. Protocols based on blind signatures hide voter's identity, but still make the actual content of a vote visible to the authority. Protocols based on blind signatures generally consist of a registration phase followed by a voting phase.

Fujioka et al. [6], developed a practical voting scheme using blind signatures. In their proposal, each voter signs his/her vote with a secret key, and then sends it to the counting center through an anonymous channel. One disadvantage of this scheme is that the protocol is complex since the voting phase consists of two steps.

In 1997, L. Cranor and R. Cytron [5] proposed and implemented a protocol based on Fujioka's scheme called *Sensus*. The main difference between both schemes is that Sensus allowed users to vote in a single session, whereas Fujioka's proposal required two sessions. However, one disadvantage of these schemes is that the network traffic

increases since the voter is required to send the same ciphered messages twice, making the protocol less efficient.

On the contrary, in Wen-Sheng et al. scheme [14], the network traffic is lower since every voter is allowed to send only a single anonymous message. Unfortunately, it has been shown that this scheme does not avoid vote duplication.

In 1998, Mu and Varadharajan [12] proposed two security schemes for electronic voting that addressed the issue of voter's privacy. Authors in [12] claimed that both protocols were capable of detecting vote duplicity. Nevertheless, in 2003 Chien et al. [15] and Lin-Hwang-Chang [1] independently found one security flaw in the Mu and Varadharajan scheme: the possibility that a user could vote more than once without being detected. Moreover, Lin et al. proposed a modification to Mu and Varadharajan's protocol, adding a protection scheme against possible frauds based on the use of blind signatures. The proposed scheme did not require any special voting channel and it was able to effectively detect vote duplicity.

Recently, however, Hwang-Wen-Hwang showed in [2] that Lin et al.'s modification is susceptible of being attacked by a corrupted Authentication Server. This attack allows the Authentication Server to identify voters of published tickets at will, thereby losing voters' privacy. Moreover, Hwang et al. proposed a modification to the Lin et al. scheme able to circumvent that attack.

Yang et al. proposed in [3] another modification to the scheme proposed by Mu and Varadharajan. That scheme resists the attacks reported in [1,15,2]. However, it seems that their scheme reported in [3] cannot obtain the identity of a malicious user trying to vote more than once.

Next Section describes Lin-Hwang-Chang [1] protocol in some detail.

3. Lin-Hwang-Chang's scheme

Lin-Hwang-Chang's scheme consists of three phases: authentication, voting and counting. In order to describe that scheme, we will use the following notation:

- V: voter's name
- AS: authentication server; VS: voting server; TCS: counting server
- Cert: digital certificate issued by a certification authority
- t: time stamp
- ||: bit concatenation
- p: large prime number
- g: a generator for Z_{p-1}^*
- $\{e_X, d_X\}$, n_X: a pair of RSA keys for entity X, where $n_X=p_1\times p_2$, p_1 and p_2 two large primes and $e_X\times d_X \bmod \phi(n_X)=1$
- $\mathrm{ENC}_X(m)$ and $\mathrm{VER}_X(m)$ denote the RSA's public operation applied over m using entity's X public key e_X, i.e, the operation of encrypting (verifying) m as, $c=m^{eX} \bmod n_X$
- $\mathrm{SIG}_X(m)$ and $\mathrm{DEC}_X(m)$ denote the RSA's private operation applied over m using entity's X private key d_X, i.e, the operation of signing (decrypting) m as, $s=m^{dX} \bmod n_X$

We will denote the modular exponentiation operation $g^e \bmod n$, as simply, g^e, whenever it is unambiguous to do so.

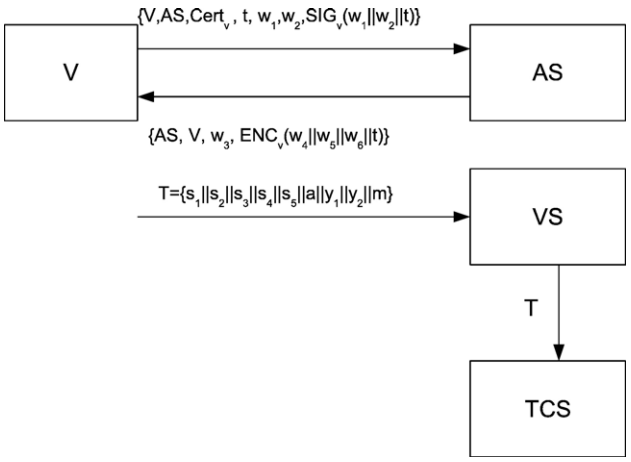

Fig. 1. Scheme proposed by Lin-Hwang-Chang.

Fig. 1 shows the dataflow of the Lin-Hwang-Chang's scheme to be described in the rest of this section.

3.1. Authentication phase

This phase consists of four steps:

(1) First the voter picks up two blind factors b_1 and b_2, along with two random numbers k_1 and r, to generate w_1 and w_2 as,

$$\begin{aligned} w_1 &= g^r \cdot \mathrm{ENC}_{\mathrm{AS}}(b_1) \\ w_2 &= g^{k_1} \cdot \mathrm{ENC}_{\mathrm{AS}}(b_2) \end{aligned} \tag{1}$$

Lastly the voter sends $\{V, \mathrm{AS}, \mathrm{Cert}_V, t, w_1, w_2, \mathrm{SIG}_V(w_1||w_2||t)\}$ to AS.

(2) AS checks the validity of both, voter's certificate and his/her signature $\mathrm{SIG}_V(w_1||w_2||t)$. If the signature is valid, AS is sure that the received parameters have not been corrupted. Then, it chooses a random number k_2 for the actual voter. Voter's identity and k_2 are stored in the AS data base. It is noticed that k_2 must be unique for each voter. Then, AS generates:

$$\begin{aligned} w_3 &= \mathrm{ENC}_V(k_2||t), \\ w_4 &= \mathrm{SIG}_{\mathrm{AS}}(w_1 \times \mathrm{AS}) = b_1 \cdot \mathrm{SIG}_{\mathrm{AS}}(a \times \mathrm{AS}), \\ w_5 &= \mathrm{SIG}_{\mathrm{AS}}(w_2 \times g^{k_2} \times \mathrm{AS}) = b_2 \cdot \mathrm{SIG}_{\mathrm{AS}}(y_1 \times \mathrm{AS}), \\ w_6 &= \mathrm{SIG}_{\mathrm{AS}}(w_2^2 \times g^{k_2} \times \mathrm{AS}) = b_2^2 \cdot \mathrm{SIG}_{\mathrm{AS}}(y_2 \times \mathrm{AS}) \end{aligned} \tag{2}$$

where $a=g^r$, $y_1=g^{k_1+k_2}$, $y_2=g^{2k_1+k_2}$. Using these values, AS responds V with the following message:

$$\{\mathrm{AS}, V, w_3, \mathrm{ENC}_V(w_4||w_5||w_6||t)\}$$

(3) V gets k_2 by decrypting w_3. In this way, V can calculate y_1 and y_2. Furthermore, V can easily obtain the signatures s_1, s_2 and s_3 by removing the blind factors as follows,

$$\begin{aligned} s_1 &= w_4 \times b_1^{-1} = \mathrm{SIG}_{\mathrm{AS}}(a \times \mathrm{AS}), \\ s_2 &= w_5 \times b_2^{-1} = \mathrm{SIG}_{\mathrm{AS}}(y_1 \times \mathrm{AS}), \\ s_3 &= w_6 \times b_2^{-2} = \mathrm{SIG}_{\mathrm{AS}}(y_2 \times \mathrm{AS}) \end{aligned} \tag{3}$$

(4) Using ElGamal digital signature, the voter proceeds to sign the voting content m, using as public keys y_1 and y_2, and as

private keys $x_1=k_1+k_2$ and $x_2=2k_1+k_2$, respectively. Each signature (a, s_4) and (a, s_5) of the vote m may be generated as,

$$\begin{aligned} s_4 &= x_1^{-1}(m\cdot \mathrm{a}-r)\bmod(p-1),\\ s_5 &= x_2^{-1}(m\cdot \mathrm{a}-r)\bmod(p-1). \end{aligned} \tag{4}$$

Finally, V can generate his voting ticket as:

$$T = \{s_1||s_2||s_3||s_4||s_5||a||y_1||y_2||m\}$$

3.2. Voting phase

This phase consists of the following steps:

(1) The voter sends his voting ticket T to the Voting Server (VS).

(2) VS verifies validity of a, y_1 and y_2 by comparing the following equations:

$$\begin{aligned} \mathrm{AS}\times a &\stackrel{?}{=} \mathrm{VER}_{\mathrm{AS}}(s_1),\\ \mathrm{AS}\times y_1 &\stackrel{?}{=} \mathrm{VER}_{\mathrm{AS}}(s_2),\\ \mathrm{AS}\times y_2 &\stackrel{?}{=} \mathrm{VER}_{\mathrm{AS}}(s_3). \end{aligned} \tag{5}$$

If all the three previous signatures verify correctly, then VS proceeds to verify the ElGamal signatures (a, s_4) and (a, s_5) of vote m using the following equations:

$$\begin{aligned} g^{ma} &\stackrel{?}{=} y_1^{s_4}\cdot \mathrm{a} \bmod p,\\ g^{ma} &\stackrel{?}{=} y_2^{s_5}\cdot \mathrm{a} \bmod p, \end{aligned}$$

respectively. If these last two verifications are satisfactory, VS can be sure of the ticket T legitimacy. VS stores all valid voting tickets that receives in order to send them later in batch to TCS over the network.

3.3. Counting phase

All voting servers send their tickets to TCS. Then, TCS publishes and counts all valid tickets received. TCS is also responsible for detecting if two or more tickets were sent by the same voter. The later is accomplished by means of the procedure depicted below.

(1) Let us assume that a voter uses the same parameters y_1, y_2 and a to generate and sign another vote m' and sent it to VS.

(2) Then, TCS will have received at least two tickets with the following form:

$$\begin{aligned} T &= \{s_1, s_2, s_3, s_4, s_5, a, y_1, y_2, m\},\\ T' &= \{s_1, s_2, s_3, s_4', s_5', a, y_1, y_2, m'\} \end{aligned}$$

(3) In this way TCS has the ability to find the identity of a malicious voter using the following equations:

$$\begin{aligned} x_1 &= \frac{m'a-ma}{s_4'-s_4}\bmod(p-1),\\ x_2 &= \frac{m'a-ma}{s_5'-s_5}\bmod(p-1),\\ k_1 &= x_2-x_1,\\ k_2 &= x_1-k_1. \end{aligned} \tag{6}$$

Finally, TCS can identify the malicious voter by searching in AS's database which voter is associated with the unique random number k_2.

3.4. Security weaknesses in Lin-Hwang-Chang's scheme

In this subsection we describe two serious weaknesses in the Lin-Hwang-Chang's scheme that could generate serious communication problems between the authentication server and the voter and compromise voter's privacy.

3.4.1. Attack from a corrupted AS

Partially based on the ideas reported in [15], Hwang et al. showed in [2] that a determined corrupted Authentication Server can utilize the TCS list of all cast votes to get the identity of any voter at will. That attack can be mounted as follows.

Let us suppose that AS wants to trace the owner of the published ticket $T=\{s_1||s_2||s_3||s_4||s_5||a||y_1||y_2||m\}$. Then AS can compute:

$$x = (y_2/y_1) = (g^{2k_1+k_2}/g^{k_1+k_2}) = g^{k_1}\bmod p. \tag{7}$$

The only remaining task for the AS is to check in its database which one of the entries $(\hat{V}, \hat{k}_2)$ satisfies the following equalities:

$$\begin{aligned} x\cdot \hat{\mathrm{g}}^{k_2} &\stackrel{?}{=} y_1\\ x^2\cdot \hat{\mathrm{g}}^{k_2} &\stackrel{?}{=} y_2 \end{aligned} \tag{8}$$

Provided that all published tickets are valid ones, this procedure will always succeed in finding out the user associated to any given vote.

In order to prevent that a corrupted AS may compute Eq. (7), Hwang et al. proposed in [2] a solution that, although technically correct, it significantly increments the overall computational cost of Lin et al. original scheme. That solution consists on working with not one but two generators selected at random in $\mathbb{Z}_{p-1}^*$, namely, g and h. Those two generators are used to create the ElGamal public keys $y_1=g^{k_1+k_2}$ and $y_2=h^{2k_1+k_2}$, respectively (see Eq. (4)). This way it becomes virtually impossible to compute Eq. (7) efficiently due to the discrete logarithm problem intractability. However, Hwang et al. solution increments by about 30% the computational effort required for computing all cryptographic operations such as signatures/verification and encryption/decryption primitives.

On the other hand, Yang et al. in [3] prevented this attack by using $y_1=g^{k_1+2k_2+2q}$ and $y_2=g^{3k_1+k_2}$ as ElGamal public keys, where q is a random number in $\mathbb{Z}_{p-1}^*$. Once again, Eq. (7) cannot be applied thus making impossible for a corrupted AS to mount this attack.

3.4.2. Unfeasibility of signing the voting content m

During Lin-Hwang-Chang's registration phase, a voter must choose 4 random numbers, among them, k_1. In addition, the authentication server selects k_2 for each valid voter. Note that the following two values x_1 and x_2 are generated between the voter and AS.

$$\begin{aligned} x_1 &= k_1 + k_2\\ x_2 &= 2k_1 + k_2. \end{aligned} \tag{9}$$

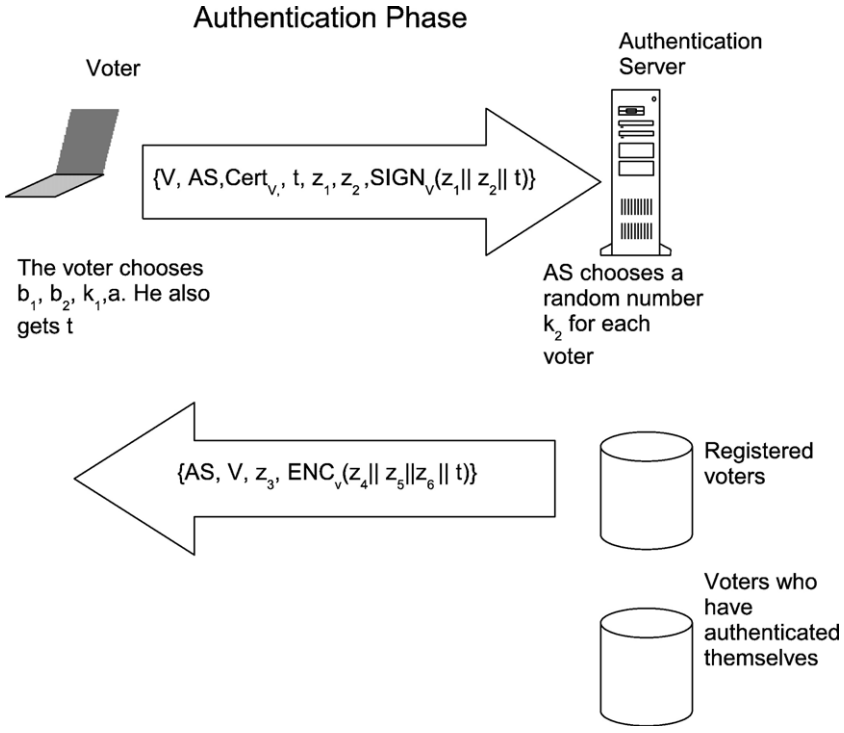

Fig. 2. First phase of the proposed scheme.

In fact, x_1 and x_2 are used by the voter as private keys in the fourth step of the authentication phase. But ElGamal signature is computed by using the following equation,

$$s = k^{-1}(m \cdot \mathrm{a} - r) \bmod (p-1) \tag{10}$$

Where k^{-1} can be computed if and only if the private key k happens to be relative prime of $(p-1)$) namely, $gcd(k, p-1) = 1$. In the case of the Lin-Hwang-Chang's scheme, a user will sign his/her vote using as private keys x_1 and x_2 (see Eq. (4)). However, given the fact that the voter selects k_1 arbitrarily, and that the authentication server does the same for the value k_2, it is perfectly possible that x_1 and x_2 may not be relative primes of $(p-1)$, thus making impossible for the voter to obtain k^{-1}.[2]

Unfortunately, this fact would not be known by the authentication server since (at least in theory) it never gets to know k_1. However, the voter will learn about this at the end of the authentication phase because he/she will not be able to generate the signatures s_4 and s_5 of Eq. (4). To complicate matters even more, the authentication server will have already associated k_2 to that voter. Thus, that user will not be able to vote since for security reasons, the authentication server cannot assign a new k_2 to him.

In the case of the Lin et al. scheme, the difficulty just outlined can be easily overcomed as follows. Let p be a prime such that $l = \frac{p-1}{2}$ is prime, thus implying that 2 and l are the solely divisors of the modulus $p-1$ in Eq. (10). Additionally, let k_1 and k_2 be even and odd integers, respectively, such that k_1, $k_2 < p/3$. Then, k_1+k_2 and $2k_1+k_2$ are both odd numbers less than l. Under these conditions, it follows that both, k_1+k_2 and $2k_1+k_2$ are coprimes with $p-1$, thus guaranteeing that a voter can always obtain k^{-1} of Eq. (10).

3.4.3. Summary

Based on above considerations we propose to generate the signatures s_4 and s_5 using the *Digital Signature Algorithm* (DSA). Let us recall that the DSA signature scheme relies on two related discrete logarithm problems. One is the logarithm problem in the field generated by a prime p. The other is the logarithm problem in the cyclic subgroup of order q, where q is a prime number such that q divides $p-1$.

As it is discussed in next Section, by using the DSA signature scheme instead of the ElGamal scheme, we can eliminate the Lin-Hwang-Chang protocol's weakness described in Section 3.4.2, without decreasing its performance and/or security.

4. Improvement to Lin-Hwang-Chang's schema

In order to substitute the ElGamal digital signature algorithm with the DSA scheme, some adjustments must be made to the voting protocol. This is because ElGamal employs mod p and mod $(p-1)$ and DSA requires mod p and mod q instead (with p and q primes). With that modification, it is guaranteed that independently of the k_1 and k_2 values that a voter and an authority server may respectively choose, a vote will be signed correctly before being sent to the voting server.

[2] Parameters k_1 and k_2 are generated exactly in the same way in Hwang et al. [2] and Yang et al. [3] security systems. In [2], $x_1 = k_1 + k_2$ and $x_2 = 2k_1 + k_2$, whereas in [3], $x_1 = k_1 + 2k_2 + q$ and $x_2 = 3k_1 + k_2$. In both cases there is a high probability that x_1 and x_2 may not be coprimes with $(p-1)$.

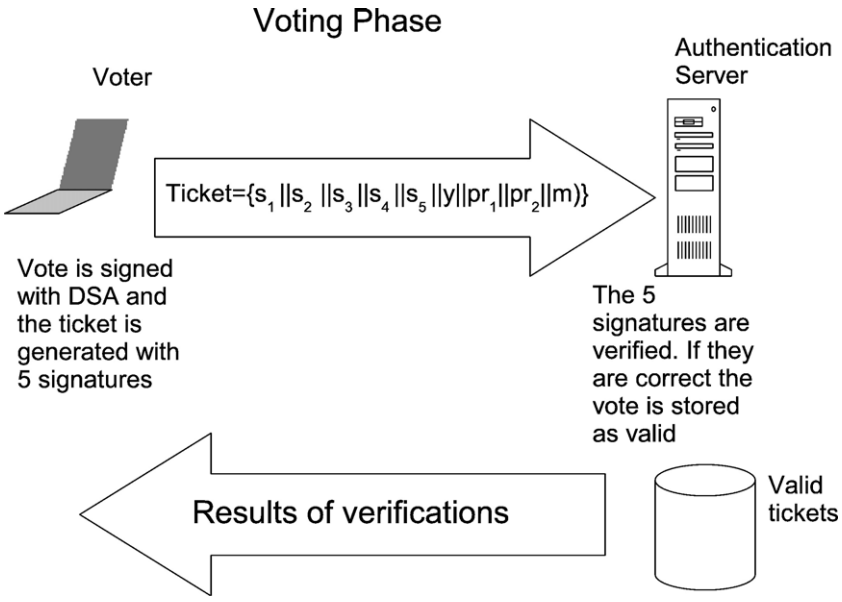

Fig. 3. Second phase of the proposed scheme.

4.1. Proposed scheme

The protocol that we propose for improving Lin-Hwang-Chang's schema contains three phases and the notation that will be used to describe its operation is as follows:

- V: voter; AS: authentication server; VS: voting server; TCS: counting server
- t: time stamp
- q: DSA parameter, $2^{159}<q<2^{160}$
- p: given l such that $0\le l\le 8$, let p be a prime such that $2^{511+64l}<p<2^{512+64l}$,with the property that q divides $(p-1)$, i.e, $q|(p-1)$.
- g: a generator for $\mathbb{Z}^*_{p-1}$
- a: DSA private key $1\le a\le q-1$
- $\alpha=g^{(p-1)/q} \bmod p$
- $y=\alpha^a \bmod p$
- Cert: digital certificate issued by an authority
- $\{e_x, d_x\}$, n_x: a pair of RSA keys for user x, where $n_x=p_1\times p_2$, p_1 and p_2 two large primes and $e_x\times d_x$, mod $\phi(n_x)=1$
- $\mathrm{ENC}_X(m)$ and $\mathrm{VER}_X(m)$ denote the RSA's public operation applied over m using entity's X public key e_X, i.e, the operation of encrypting (verifying) m as, $c=m^{eX} \bmod n_X$
- $\mathrm{SIG}_X(m)$ and $\mathrm{DEC}_X(m)$ denote the RSA's private operation applied over m using entity's X private key d_X, i.e, the operation of signing (decrypting) m as, $s=m^{dX} \bmod n_X$.

Once again, we will denote the modular exponentiation operation $g^e \bmod n$, as simply, g^e, whenever it is unambiguous to do so.

4.2. Authentication

The authentication phase of the proposed protocol is shown in Fig. 2. This phase contains three steps:

(1) The voter chooses two blind factors $b_1<n_{\mathrm{AS}}$ and $b_2<n_{\mathrm{AS}}$ and two random numbers $k_1<\frac{q}{3}$ and a in $\mathbb{Z}^*_{q-1}$. It is noticed that a will be used as Voter's DSA private key. Using these values together with the DSA public parameters, the values y, z_1 and z_2 are generated in the following way:

$$\begin{aligned} y &= \alpha^a \bmod p,\\ z_1 &= y\cdot \mathrm{ENC}_{\mathrm{AS}}(b_1),\\ z_2 &= (\alpha^{k_1} \bmod p)\cdot \mathrm{ENC}_{\mathrm{AS}}(b_2). \end{aligned} \tag{11}$$

where p and α are DSA public domain parameters.

Then the voter sends $\{V, \mathrm{AS}, \mathrm{Cert}_V, t, z_1, z_2, \mathrm{SIGN}_V(z_1||z_2||t)\}$ to AS.

(2) AS validates V's identity by verifying both, the received signature $\mathrm{S1GN}_V(z_1||z_2||t)$ and the public key included in Cert_V. If the signature is valid, AS chooses a random number $k_2<\frac{q}{3}$ and stores it in the database as an identification of V. For this reason the value k_2 must be unique for each voter. Then AS generates z_3, z_4, z_5 and z_6 using the following equations:

$$\begin{aligned} z_3 &= \mathrm{ENC}_V(k_2||t),\\ z_4 &= \mathrm{SIG}_{\mathrm{AS}}(z_1\times \mathrm{AS}) = b_1\cdot \mathrm{SIG}_{\mathrm{AS}}(y\times \mathrm{AS})\\ z_5 &= \mathrm{SIG}_{\mathrm{AS}}(z_2\times(\alpha^{k_2} \bmod p)\times \mathrm{AS})\\ &= b_2\cdot \mathrm{SIG}_{\mathrm{AS}}(\alpha^{k_1+k_2} \bmod p\times \mathrm{AS})\\ z_6 &= \mathrm{SIG}_{\mathrm{AS}}(z_2^2\times(\alpha^{k_2} \bmod p)\times \mathrm{AS})\\ &= b_2^2\cdot \mathrm{SIG}_{\mathrm{AS}}(\alpha^{2k_1+k_2} \bmod p\times \mathrm{AS}) \end{aligned} \tag{12}$$

Finally, AS sends the following reply message to V,

$$\{\mathrm{AS}, V, z_3, \mathrm{ENC}_V(z_4||z_5||z_6||t)\}$$

Notice that in this message the values z_4, z_5 and z_6 are encrypted with V's public key. Additionally, a timestamp t is added to the message.

(3) The voter decrypts z_3 to get k_2. Additionally, he/she decrypts z_4, z_5 and z_6 using his/her private exponent d_V. Then,

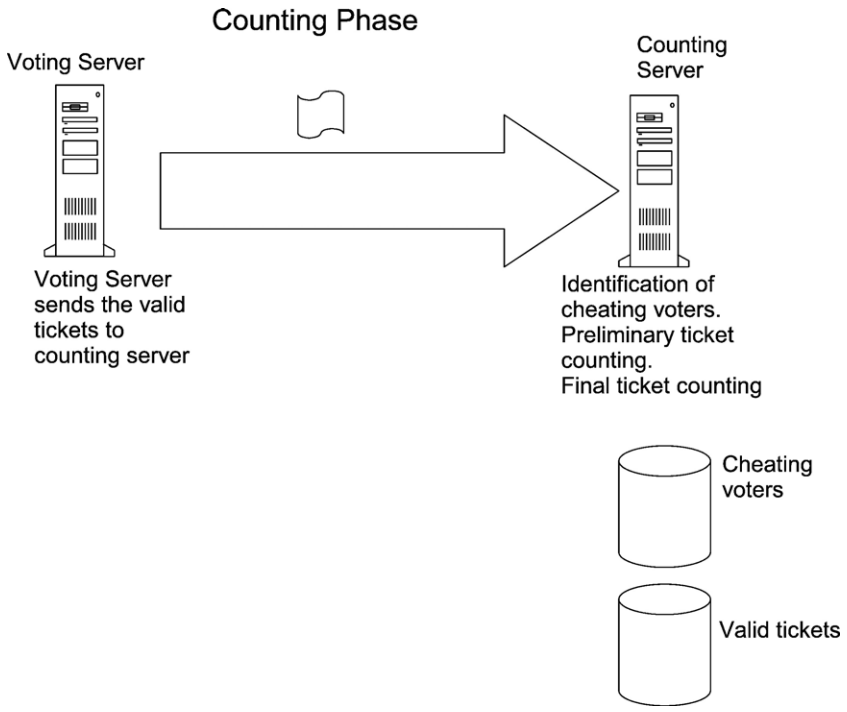

Fig. 4. Third phase of the proposed scheme.

the blind factors are removed so that the signatures s_1, s_2 and s_3 can be obtained as follows,

$$\begin{aligned} s_1 &= z_4 \times b_1^{-1} = \mathrm{SIG}_{\mathrm{AS}}(y \times \mathrm{AS}) \\ s_2 &= z_5 \times b_2^{-1} = \mathrm{SIG}_{\mathrm{AS}}(\alpha^{k_1+k_2} \bmod p \times \mathrm{AS}) \\ s_3 &= z_6 \times b_2^{-2} = \mathrm{SIG}_{\mathrm{AS}}(\alpha^{2k_1+k_2} \bmod p \times \mathrm{AS}) \end{aligned} \tag{13}$$

4.3. Voting phase

The dataflow of the proposed protocol voting phase is shown in Fig. 3.

(1) In the voting phase the voter proceeds to sign the ballot (m) using the DSA scheme and \$a\$, x_1 and x_2 as private keys. The voter is able to generate these values because he/she has already decrypted k_2. Notice that the two DSA signatures consists on the pairs (r_1, s_4) and (r_2, s_5). Where the first component of each signature, namely r_1 and r_2, can be obtained as follows,

$$\begin{aligned} x_1 &= k_1 + k_2, \\ x_2 &= 2k_1 + k_2, \\ r_1 &= (\alpha^{x_1} \bmod p) \bmod q, \\ r_2 &= (\alpha^{x_2} \bmod p) \bmod q. \end{aligned} \tag{14}$$

Whereas the second component of the DSA signatures, namely s_4 and s_5, can be generated through the computation of

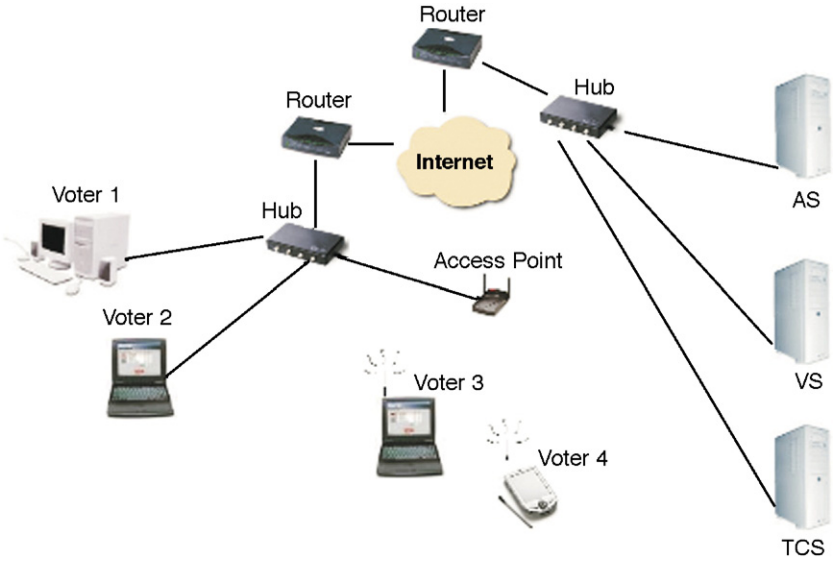

Fig. 5. Testing system for the e-voting protocol.

Table 1
Cryptographic operations

PHaSE	Voter	Authorized service	Voting service
Authentication	1 RSA signature 2 RSA encryptions	1 RSA verification 4 RSA encryptions 3 RSA blind signatures	
Voting	4 RSA decryptions 2 DSA signatures		3 RSA verifications 2 DSA verifications
TOTAL	9 operations	8 operations	5 operations

Table 3
Comparison table

Scheme	RSA operations	ElGamal operations	DSA operations
Lin et al. [1]	10 public ops+8 priv ops	2 public ops+2 priv ops	None
Hwang et al. [2]	14 public ops+10 priv ops	2 public ops+2 priv ops	None
Yang et al. [3]	8 public ops+6 priv ops	2 public ops+2 priv ops	None
This Protocol	10 public ops+8 priv ops	None	2 public ops+2 priv ops

the following equations:

$$s_4 = x_1^{-1}(m + ar_1) \bmod q,$$
$$s_5 = x_2^{-1}(m + ar_2) \bmod q. \tag{15}$$

Additionally the voter must compute the values l_1 and l_2 defined as,

$$l_1 = [((\alpha^{k_1} \bmod p) \bmod n_{AS}) \times ((\alpha^{k_2} \bmod p) \bmod n_{AS})] \bmod n_{AS},$$
$$l_2 = [((\alpha^{k_1} \bmod p)^2 \bmod n_{AS}) \times ((\alpha^{k_2} \bmod p) \bmod n_{AS})] \bmod n_{AS}. \tag{16}$$

These last two values together with r_1 and r_2 are encapsulated taking advantage of the *Chinese Residue Theorem*. That is done with the goal of allowing VS to perform the corresponding verifications in the proper arithmetic (either modulus n_{AS} or modulus q) and also in order to keep the size of the vote as small as possible.

$$pr_1 = [(r_1 \times n_{AS}) + (l_1 \times q)] \bmod (n_{AS} \cdot \mathrm{q})$$
$$pr_2 = [(r_2 \times n_{AS}) + (l_2 \times q)] \bmod (n_{AS} \cdot \mathrm{q}) \tag{17}$$

Lastly, the voting ticket is generated in the following way:

$$\text{Ticket} = \{s_1, s_2, s_3, s_4, s_5, y, pr_1, pr_2, m\}$$

(2) The voter sends his voting ticket to VS. VS performs a total of 5 signature verifications needed for ticket validation. The first 3 verification equations are as follows

$$(\mathrm{AS} \times y) \bmod n_{AS} \stackrel{?}{=} \mathrm{VER}_{AS}(s_1),$$
$$(\mathrm{AS} \times pr_1 \cdot q^{-1}) \bmod n_{AS} \stackrel{?}{=} \mathrm{VER}_{AS}(s_2), \tag{18}$$
$$(\mathrm{AS} \times pr_2 \cdot q^{-1}) \bmod n_{AS} \stackrel{?}{=} \mathrm{VER}_{AS}(s_3)$$

Notice that in virtue of the *Chinese Residue Theorem* we have that $pr_1 \cdot q^{-1} = l_1$ and $pr_2 \cdot q^{-1} = l_2$, where q^{-1} is defined as the multiplicative inverse of q modulus n_{AS}. Similarly r_1 and r_2 can be recovered by computing $pr_1 \cdot n_{AS}^{-1} = r_1$ and $pr_2 \cdot n_{AS}^{-1} = r_2$, where n_{AS}^{-1} is defined as the multiplicative inverse of n_{AS} modulus q.

Table 2
Approximate size of messages

Phase	Message 1	Message 2
Authentication	19.5 Kb	4.5 Kb
Voting	6.7 Kb	–

We verify the DSA signatures using the standard procedure shown below,

DSA_Verification(r, s) {
1. Check whether $0<r<q$ and $0<s<q$
2. $w = s^{-1} \bmod q$
3. $u_1 = w \cdot m \bmod q$
4. $u_2 = r \cdot w \bmod q$
5. $v = (\alpha^{u1} y^{u2} \bmod p) \bmod q$
6. return v}

Then the last two signatures s_4 and s_5 can be verified as follows,

$$r_1 \stackrel{?}{=} \mathrm{DSAverifier}(r_1, s_4)$$
$$r_2 \stackrel{?}{=} \mathrm{DSAverifier}(r_2, s_5) \tag{19}$$

(3) If all five signatures are correctly verified, VS will accept and store the ticket sent by the voter as a valid one. Once that the voting election process has been completed VS sends all valid votes that were received to TCS over the communication network.

4.4. Counting phase

TCS must receive all valid tickets from the voting servers. Additionally, TCS must identify all tickets that are identical and count them only once. These actions will guarantee a final tally equal to the total number of the valid votes received during the elections.

In this phase it is possible to detect malicious voters that may have sent two or more tickets with different votes. In order to perform the so-called *double voting detection*, we consider the scenario where a given voter uses the same key to sign different votes. Therefore, TCS will receive at least two tickets with the following form:

$$B_1 = \{s_1, s_2, s_3, s_4, s_5, y, pr_1, pr_2, m\},$$

$$B_2 = \{s_1, s_2, s_3, s_4', s_5', y, pr_1, pr_2, m'\}.$$

With the information contained in these two tickets, TCS is capable of identifying the voter who sent these ballots, by computing the following equations:

$$x_1 = \frac{m' - m}{s_4' - s_4} \bmod q,$$

$$x_2 = \frac{m' - m}{s_5' - s_5} \bmod q,$$

$$k_1 = x_2 - x_1,$$

$$k_2 = x_1 - k_1.$$

As it was mentioned previously, all k_2 values assigned to each voter are stored in the database of AS. In this way TCS can request to AS the name of the voter which is associated to the computed value k_2, thus identifying the identity of the malicious voter. The flowchart of this phase is shown in Fig. 4.

4.5. Security analysis of the proposed solution

Much of the security properties of the Lin et al. scheme apply to our proposed DSA solution. Moreover, the functional flaw described in Section 3.4.2 has been effectively removed. To see this, notice that in our solution, the ElGamal signing Eq. (10) of Lin et al. scheme has been substituted by the DSA signing Eq. (15). The existence of the multiplicative inverses x_1^{-1} and x_2^{-1} is always guaranteed due to the fact that the modulus in Eq. (15) is the DSA parameter q, which happens to be a prime number. Thus both, $x_1 < q$ and $x_2 < q$ must be coprimes to q.[3]

Certainly, the DSA/RSA based scheme described in Section 4 can still be attacked by a corrupted AS as described in Section 3.4.1. However, our DSA solution can be adapted to be used in both, Hwang et al. and Yang et al. schemes, thus preventing that attack.

5. Implementation

We implemented the modifications to the Lin-Hwang-Chang's scheme in a fully functional e-voting system to test its performance. Fig. 5 shows the configuration of the testing system.

The cryptographic operations described in the protocol were implemented with the standard cryptographic libraries available in Java's language SDK. The authority servers and the voter application were implemented with servlets/jsp and applets, respectively. A client application was also written for voters with mobile devices such PDAs. We also employed Apache an standard web server using Tomcat as servlet container. All voting data from the elections was stored in MySQL DBMS.

In our experiments, the e-voting system was able to obtain a final exact tally of up to five thousand votes in less than 140 s.

[3] Let us recall that since $k_1 < \frac{q}{3}$ and $k_2 < \frac{q}{3}$, it implies that $x_1 = k_1 + k_2 < q$ and $x_2 = 2k_1 + k_2 < q$.

5.1. Keys and certificates

As illustrated in Fig. 5, the authorities involved in our scheme are: Authentication Server (AS), Voting Server (VS) and Counting Server (TCS). Additionally, a certification authority is also needed. This authority generates the pair of RSA keys with 1024 bits and the digital certificate corresponding to the public key used in the phase of voting. This last certificate is needed by a voter to authenticate him/herself with the AS during an election.

For testing purposes, a digital certification authority was created for voter's client applications. Hence, during the key generation process performed by the certification authority, the client must provide a password, which will be used for recovering and decrypting the private key. This is done applying the hash function MD5 together with the DES block cipher algorithm. Subsequently, the client gets a public key, a digital certificate and a certified private key.

6. Evaluation and comparative analysis

Our protocol was evaluated in terms of the number of cryptographic operations performed in the different phases. The number of cryptographic operations performed by the protocol during the authentication and voting processes is shown in Table 1.

The size and number of the messages that are transferred during the authentication and voting phases are shown in Table 2.

Table 3 shows a comparison of our protocol with the protocols reported [1–3], in terms of the total number of cryptographic operations

7. Conclusion

We have presented an efficient and effective e-voting protocol for secure on-line elections. Our proposed protocol includes a new scheme that corrects a flaw found in both Lin-Hwang-Chang's scheme [1] and a recent modification of it by Hwang-Wen-Hwang [2]. Moreover, we found that the more recent scheme proposed by Yang-Lin-Yang [3] also suffers from this same problem. Our scheme substitutes the ElGamal digital signatures employed by the other protocols with RSA/DSA, guaranteeing that independently of the values k_1 and k_2 chosen by the voter and authentication server, all required signatures can be always generated by a voter. This occurs since it holds that the parameters $k_1 + k_2$ and $2k_1 + k_2$ and the DSA public parameter q, are coprimes.

We have implemented the protocol and measured its performance in terms of number of cryptographic operations executed and messages sent in the different operational phases. Our performance results show that the protocol performs comparatively well with other protocols and is capable of handling a few thousand votes efficiently using standard middleware components in a multi-tier architecture.

References

[1] I. Lin, M. Hwang, C. Chang, Security enhancement for anonymous secure e-voting over a network, Comput. Stand. Interfaces 25 (2) (2003) 131–139.

[2] S. Hwang, H. Wen, T. Hwang, On the security enhancement for anonymous secure e-voting over computer network, Comput. Stand. Interfaces 27 (2) (2005) 163–168.
[3] C. Yang, C. Lin, H. Yang, Improved anonymous secure e-voting over a network, Inf. Secur. 15 (2) (2004) 185–191.
[4] D. Chaum, Blind signatures for untraceable payments, CRYPTO, 1982, pp. 199–203.
[5] L. Cranor, R. Cytron, Sensus: a security-conscious electronic polling system for the Internet, HICSS, vol. 3, 1997, pp. 561–570.
[6] A. Fujioka, T. Okamoto, K. Ohta, A practical secret voting scheme for large scale elections, ASIACRYPT '92: Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques, Springer-Verlag, London, UK, 1993, pp. 244–251.
[7] R. Joaquim, A. Zuquete, P. Ferreira, Revs — a robust electronic voting system, Proceedings of IADIS International Conference e-Society, 2003, pp. 95–103.
[8] J. Karro, J. Wang, Towards a practical, secure, and very large scale online election, ACSAC '99: Proceedings of the 15th Annual Computer Security Applications Conference, IEEE Computer Society, Washington, DC, USA, 1999, p. 161.
[9] I. Ray, I. Ray, N. Narasimhamurthi, An anonymous electronic voting protocol for voting over the Internet, WECWIS '01: Proceedings of the Third International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS '01), IEEE Computer Society, Washington, DC, USA, 2001, p. 188.
[10] B. Schoenmakers, A simple publicly verifiable secret sharing scheme and its application to electronic, CRYPTO '99: Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology, Springer-Verlag, London, UK, 1999, pp. 148–164.
[11] K. Iversen, A cryptographic scheme for computerized elections, Advances in Cryptology — CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11–15, Vol. 576 of Lecture Notes in Computer Science, Springer, 1992, pp. 405–419.
[12] Y. Mu, V. Varadharajan, Anonymous secure e-voting over a network, ACSAC '98: Proceedings of the 14th Annual Computer Security Applications Conference, IEEE Computer Society, Washington, DC, USA, 1998, p. 293.
[13] FIPS 186-2, Digital Signature Standard, National Institute of Standards and Technology (NIST), http://csrc.nist.gov/publications/fips/October 2001.
[14] W. Juang, C. Lei, A secure and practical electronic voting scheme for real enviroments, IEICE Trans. Fundam. E80-A (1) (1997) 64–71.
[15] H. Chien, J. Jan, Y. Tseng, Cryptanalysis on Mu–Varadharajan's e-voting schemes, Appl. Math. Comput. 139 (2–3) (2003) 525–530.

Francisco Rodríguez-Henríquez received the PhD (2000) degree in electrical and computer engineering from Oregon State University, the M.Sc. (1992) degree in electrical and computer engineering from the National Institute of Astrophysics, Optics and Electronics (INAOE), México and the B.Sc. (1989) degree in electrical engineering from the University of Puebla, México. Currently, he is an associate professor at the Computer Science Department of the Advanced Research Center (CINVESTAV-IPN), in Mexico City, México, which he joined in 2002. Dr. Rodríguez-Henríquez has co-authored over 40 technical papers. He has also co-authored the book *Cryptographic Algorithms on Reconfigurable Hardware* (Springer, November 2006). His major research interests are in data security, cryptography, finite fields, error correcting codes, and mobile computing. He is a member of the IEEE and he is also an alumni member and research associate of the Information Security Laboratory at Oregon State University.

Daniel Ortiz-Arroyo received a PhD degree in computer engineering from the School of Electrical Engineering and Computer Science at Oregon State University and the M.Sc. degree in computer engineering from the Electronics Department at National Institute of Astrophysics, Optics and Electronics (INAOE), México. In 2003 he joined the Computer Science Department at Aalborg University Esbjerg in Denmark where he is currently an associate professor. Before joining Aalborg University Esbjerg he was director of software development for a startup company in USA and professor in the Electronics Department at INAOE and in the Autonomous University of Puebla in México. He has co-authored over 30 technical papers and research monographs and edited conference proceedings. His main research interests are in computer architecture, data security, soft computing, machine learning, information retrieval and mobile computing.

Claudia García-Zamora received the M.Sc. (2005) degree in electrical and computer engineering from the Advanced Research Center (CINVESTAVIPN), in Mexico City, México. Her major research interests are in data security and cryptography.