Identity based Threshold Cryptography and Blind Signatures for Electronic Voting

GINA GALLEGOS-GARCÍA[1], ROBERTO GÓMEZ-CÁRDENAS[2], GONZALO I. DUCHÉN-SÁNCHEZ[1]
[1]Graduate and Research Section, [2]Department of Computer Science
[1]Instituto Politécnico Nacional, [2]Instituto Tecnológico de Estudios Superiores de Monterrey-CEM
[1] Av. Sta. Ana 1000, Sn. Fco. Culhuacan, 04430, Coyoacán
[1]Mexico City
[2] Carretera Lago de Guadalupe Km 3.5, Atizapán de Zaragoza, 52926
[2]Mexico State
gina@calmecac.esimecu.ipn.mx

Abstract: - Recently, there has been an increasing interest to improve the efficiency in election processes which has brought as a consequence a wide range of proposals for electronic voting. Electronic voting protocols are a reasonable alternative to conventional elections. Nevertheless, they are facing an evolution due to its requirements, especially the ones needed to provide full security considered to represent a democratic electronic vote. In the literature, different protocols based on public key schemes have been proposed to meet such security requirements. In this paper, we propose the use of bilinear pairings in order to provide the security requirements that an electronic voting protocol must meet, without requiring the entire infrastructure needed in a public key scheme. Proposed protocol considers two cryptographic primitives as main building blocks: threshold and blind signature schemes. It is divided in four main stages: set-up, authentication, voting and counting. Moreover, it meets privacy, accuracy and robustness by using bilinear pairings. We make a comparative analysis, which is based on its performance and the key pairs, Trust and Certification Authorities it requires.

Key-Words: - Bilinear pairings, Blind signatures, Electronic voting protocols, Identity based cryptography, Public key cryptography, Security requirements, Threshold cryptography.

1 Introduction

Electronic voting, as an e-government issue [1], has been mentioned in different media as the use of computers or computerized voting equipment to cast ballots in an election since 1964, which nowadays is a reasonable alternative to conventional elections and other opinion expressing processes. However, it must offer at least as same benefits as a conventional election does, in addition to reduce monetary costs.

Generally speaking an electronic voting protocol involves three main entities: voter, registration authorities and tallying authorities. The voter is an entity who has the right for voting. The registration authorities register voters before the election´s day and they also ensure only registered voters will be able to vote. The tallying authorities ensure cast votes are counted.

All those actors each other interact during three main phases: registration, voting and counting. In the registration phase, a citizen must be registered as an authenticated voter. In the voting phase, only authenticated voters cast their votes. Finally in the counting phase, performed by tallying authorities or a special center, cast votes are counted and tally is published.

In order to use an electronic voting protocol inside an electronic voting process, it must meet at least seven security requirements: privacy, eligibility, uniqueness, uncoercibility, transparency, accuracy and robustness.

Privacy: A vote must be kept secret from any coalition of authorities.

Eligibility: Only registered voters, who meet certain pre-determined criterion, must are eligible to vote.

Uniqueness: Only one vote for a voter must be counted.

Uncoercibility: Any coercers, even authorities, must be able to coercer a voter to cast its vote in a particular way.

Transparency: The whole voting protocol must be transparent, since the beginning until the end.

Accuracy: The votes should be correctly recorded and tallied.

Robustness: The protocol will be able to tolerate some faulty authorities who try to cheat during the computation of the tally.

Roughly speaking electronic voting protocols in the literature can be classified into three basic types: protocols based on mix-nets [2], protocols based on blind signatures [3] and protocols based on threshold cryptography [4]. All of them are based on Public Key Cryptography PKC, which offers high flexibility through key agreement protocols and authentication mechanisms. However, when PKC is used, it is necessary to implement a Public Key Infrastructure PKI [5] to provide certificates which bind public keys to entities, and enable other ones to verify such public key bindings. As a consequence of this, the components of the protocol increase notably.

An alternative to the use of a PKI is the Identity Based Cryptography (IBC), also named Identity Based Public Key Cryptography (ID-PKC). With this kind of cryptography, it is possible to have all the benefits offered by PKC, but without the need of certificates and nor all the core components of a PKI infrastructure.

In a cryptosystem based on identity, the public key is retrieved from an identity of the entity, and the private key is securely distributed by a Key Distribution Center.

IB-PKC was proposed by Shamir in 1984 [6], but the first practical implementation was made by Boneh in 2001[7].

Most common IB-PKC implementations are based on bilinear pairings, which have been widely studied in order to propose different protocols [8].

In this paper, we propose the use of threshold and blind signature schemes from bilinear pairings into electronic voting protocols, to ensure the security requirements such protocols must meet, without the use of certificates and their management neither a PKI infrastructure. In order to compare our work with previous proposals, we present the security requirements our protocol meets and a performance comparison based on cryptographic operations, key pairs and authorities required in our protocol.

The remainder of this document is organized as follows. In Section 2 Identity Based Public Key Cryptography, blind signatures and threshold schemes are summarized. Related work is presented in Section 3. Section 4 introduces our electronic voting protocol. Section 5 shows the comparative analysis of our protocol, which is made from two points of view: security requirements and performance comparison. Finally, Section 6 presents our conclusions and draft further work for this research.

2 Preliminaries

As mentioned before, the cryptographic primitives we use are from bilinear pairings, first practical implementation of Identity Based Public Key Cryptography. The encryption primitive, Identity Based Encryption, considers a threshold scheme. The signature primitive uses blind signatures schemes. Considering the aforementioned, we give some brief definitions about it.

2.1 Identity Based Public Key Cryptography

In the middle 70´s, Diffie *et al* introduced the asymmetric cryptography concept [9], which considers the generation of a key pair. The owner of this pair retains the one half of the key pair for private use, while allowing the other half to be made public. Asymmetric cryptography has two variants: the Public Key Cryptography PKC and the Identity Based Public Key Cryptography IB-PKC.

The central difference between both of them, and in which we are interested, is in the generation of the keys. In PKC the key pair is generated from random information unrelated to the method identifying such a key pair. Consequently, there is a requirement for a certificate to bind the public key to its main use.

Nowadays, the primary mean of deploying PKC is the Public Key Infrastructure, PKI. With this, it is necessary to have all the core components of a PKI [5]. However, the difficult inherent in running a PKI is in the managing of the certificates and associated key.

Shamir was the first person to propose a concept as a means of overcoming this problem; it was named Identity Based Public Key Cryptography, ID-PKC. In ID-PKC the key pair is generated unequivocally from data that are of relevance to the usage of the key. With this, user's identifier information such as: e-mail, IP address or serial numbers can be used as public key for encryption or signature verification. As a result, ID-PKC significantly reduces the system complexity and the cost for establishing and managing public keys authentication framework known as PKI.

The proposed concept of Shamir became a long-lasting open problem from the Identity Based Encryption's point of view, which until 2001 was solved by Boneh *et al* [7] and by Cocks [10]. Thanks to their successful realization, ID-PKC is now used as a mean to design new cryptographic protocols [8].

2.2 Blind Signatures Schemes

The blind signature schemes have the particular characteristic that neither the signers do not know the content of the message to be signed, nor the signatures that the recipients obtain for their message. This kind of signatures is used in scenarios where the signer and the message creator are different entities. The first construction of cryptographic blind signature was proposed by Chaum in [11]. Recently blind signature schemes and Identity Based Blind Signature schemes were proposed in [12] [13].

2.3 Threshold Schemes

The *(t,n)* - threshold scheme [14] considers secret information denoted by "*s*", which is shared with the help of a Private Key Generator, PKG. Such an "*s*" is not revealed unless any of *t* out of *n* participants, or shareholders, work together to reconstruct it.

A threshold scheme based on Lagrange interpolation was developed by Shamir [15], which is divided in two phases: distribution and re-construction. The basic idea, in the distribution phase, is as follows.

Let q be a prime. The information $s \in Z_q^*$ generated by PKG, should be distributed. There is a group of n members G_i *(i=1,2,3,…,n).* Fisrt, PKG randomly chooses $a_1, \dots, a_{t-1} \in Z_q^*$ and forms a distribution function $f(x) = s + a_1 x + \cdots + a_{t-1}x^{t-1}$. Then, PKG computes $x_i = f(i) \in Z_q^*$ and sends (i, x_i) to each member G_i. It is possible to note that when $i = 0$, we can obtain the information $s = x_0 = f(0)$.

In the re-construction phase it is necessary to use the Lagrange interpolation coefficient. The idea is the following: Let $S \subseteq \{1, \dots, n\}$ be a set such that $|S| \geq t$, where $|S|$ denotes the cardinality of a given set. The function $f(x)$ can be reconstructed by computing Equation (1):

$$f(x) = \sum_{j \in S} L_j x_j \tag{1}$$

with

$$L_j = \prod_{\substack{j \in S \\ j \neq i}} \frac{-x_j}{(x_i - x_j)} \tag{2}$$

Where $L_j \in Z_q^*$ is the Lagrange interpolation coefficient used in Shamir's secret sharing scheme.

Recently, Identity Based Threshold Decryption schemes were proposed in [16] [17]. In such schemes the PKG generates its public and private key, named master public key and master private key, which is no revealed. However, any encrypted text, with the identifier information of any of the *n* entities, is only decrypted with certain number of decryption shares, provided from *t* shares of the master private key.

2.4 Bilinear Pairings

Since our protocol uses schemes from bilinear pairings on elliptic curves, we give some brief definitions on their properties and their complexity assumptions.

To do this, the following statements are considered:

- There is an additive group $\mathbb{G}_1$ with ∞ as identity element. This group defines the group of points of the elliptic curve E with:

$$E(K): y^2 + ay = x^3 + bx^2 + cx + d \tag{3}$$

where E is defined over a finite field $K=GF(p^m)$ with $a, b, c, d \in K$ and p is prime.

- Each point in the elliptic curve is denoted with capital letter *P*, and the scalar multiplication of such a point is denoted by *aP*.

- There is a multiplicative group $\mathbb{G}_2$ with identity 1.

- $\mathbb{G}_1$ and $\mathbb{G}_2$ are cyclic groups of order prime *q*.

Considering the aforementioned, a bilinear pairing on $(\mathbb{G}_1, \mathbb{G}_2)$, is a map $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, that satisfies the following properties:

1. Bilinearity: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1$ and $a, b \in \mathrm{Z}_q^*$.
2. Non/degeneracy: If P is a generator of $\mathbb{G}_1$, then $\hat{e}(P, P)$ is a generator of $\mathbb{G}_2$. In other words, $\hat{e}(P, P) = g$ with $g \in GF(p^m)^k$ and k denotes the embedding degree of the curve.
3. Computable: There exists an efficient algorithm to compute $\hat{e}(P, Q)$ for all $P, Q \in \mathbb{G}_1$. Examples of cryptographic bilinear pairings are the modified Weil pairing and Tate pairing [7] [18].

With such group $\mathbb{G}_1$, we can define the following hard cryptographic problems:

- Discrete Logarithm Problem (DLP): Given $P, P' \in \mathbb{G}_1$, find an integer n such that $P = nP'$ whenever such integer exists.
- Computational Diffie-Hellman Problem (CDHP): Given a triple $(P, aP, bP) \in \mathbb{G}_1$ for $a, b \in \mathrm{Z}_q^*$ find the element abP.
- Decision Diffie-Hellman Problem (DDHP): Given a quadruple $(P, aP, bP, cP) \in \mathbb{G}_1$ for $a, b, c \in \mathrm{Z}_q^*$, decide whether $c \equiv ab(mod q)$ or not.

We assume through this paper that CDHP and DLP are intractable, which means there does not exist polynomial time algorithm to solve CDHP or DLP with non-negligible probability. When the DDHP is easy but the CDHP is hard on the group *G*, we call *G* a Gap Diffie-Hellman (GDH) group. Such groups can be found on supersingular elliptic curves or hyperelliptic curves over finite fields. Our electronic voting protocol can be built on any GDH group.

3 Related Work

In [19], a protocol based on a threshold encryption scheme, a digital signature scheme and a blind signature scheme, is proposed. In their protocol the voters do not need to join to the counting stage, hence the voters can walk away once they cast their ballots.

The Cramer *et al* proposal [4] employs a fault-tolerant threshold cryptosystem. The protocol provides the voters a public key to encrypt their votes. The corresponding private key is shared among the authorities using threshold cryptographic techniques. The private key is used implicitly when the authorities cooperate to decrypt the final tally.

Baudron *et al* [20] propose a voting protocol that guarantees privacy of voters, public verifiability and robustness against a coalition of malicious authorities. Furthermore, they address the problem of free receipt and uncoercibility of voters. All of this is achieved by using the Pailier cryptosystem [21] and zero-acknowledge proof techniques. It is a large group-oriented system, because the election organization of this proposal is divided in levels: local center level, regional level and national level.

In [22] Mu *et al* presents a protocol based on ElGamal digital signature algorithm. In these protocols users in the system, share a public key, while the signer has a secret key which is used to sign the vote.

In [23], Gallegos *et al* propose the first protocol based on threshold identity-based cryptography. It considers a responsibility distributed model, in which the votes are decrypted with *t* of *n* users.

The proposed protocols in [19] [20] [22] are based on Public Key Cryptography. As mentioned before, when PKC is used, it is necessary running a Public Key Infrastructure to manage all the certificates needed to verify the public key owner's identity. As a consequence, the cost and complexity of the PKI infrastructure makes it difficult to integrate in electronic vote protocols. Moreover, the proposed protocol in [23] has the disadvantage that shared private keys of the voters are generated by a Private Key Generator, PKG, who could act as a malicious entity and break the protocol. We improve [19] [20] [22] by eliminating the use of a PKI. Moreover, we eliminate the PKG used in [23] in order to improve mentioned disadvantage.

The entities described in Section 1 will create the public and private keys used to encrypt and decrypt the votes.

4 Protocol Description

4.3 Our electronic voting protocol

Our proposal is based on two cryptographic primitives, the threshold version [17] of the Identity Based Encryption scheme proposed in [7] and the blind signature scheme proposed in [12]. In [17] all the parameters required to produce the key pairs used in the protocol are generated by a Private Key Generator, PKG. Considering the idea proposed in [24] we decided not to use a PKG in order to generate this parameters. Instead, all the participating entities exchange information in order to produce a master public key and it corresponding master private key.

The electronic voting protocol is divided in four phases which are explained in the following sections.

4.3.1 Voting Set-Up

This stage generates the key pairs to be used in the encryption and signature cryptographic primitives. A first key pair $< Pr, Pu >$ is used to encrypt the votes with the public key Pu and decrypt them with its respective private key Pr at the counting phase. The generation of this key pair involves the participation of n entities or shareholders, E_i, where $1 \leq i \leq n$. These entities are composed by the President of the Ballot Box, the representative of the political parties, some civilians, officials and a federal authority. Each entity broadcasts and receives specific information by using a secret-sharing technique in order to generates its private share s_i, and its public share q_i. Public share is used to generate public key Pu used by the voters, to encrypt the votes during the voting stage. Private share is used during the counting stage to generate private key Pr in order to decrypt the votes. With the intention of guarantee anyone be able to send false information, private and public shares must be kept in secret.

Another key pair generated in this stage is the President of Ballot Box's private/public key pair, PBB_s and PBB_p respectively. They are used to blindly sign. PBB_s is used in the voting stage to produce a blind signature and PBB_p is sent to the Combining Entity, *CE*, which is in charge of verifying the signatures.

4.3.2 Authentication

The authentication stage is performed by asking each voter to show its identity card and checking if its name appears on a list. This stage is performed by officials and all the voters.

4.3.3 Voting

Given the identity *ID* of any entity E_i, the voter selects a candidate, which is encrypted by using public key Pu and such entity's public key, denoted by Q_{ID}. Then, it is blindly signed with PBB_s. The signed and encrypted vote is sent to all entities E_i. A hash value, obtained by using the signed and encrypted vote and a timestamp, is delivered to the voter as a receipt. Finally the identity card is marked, so the voter cannot vote more than once.

4.3.4 Counting

In this stage, the votes are verified, decrypted and counted by the Combining Entity, *CE*, who is in charge of verifying the signature and decrypting the votes. The signatures of the votes are verified with President of the Ballot Box's public key, PBB_p, and with the intention of decrypt the votes, the *CE* selects t of n decryption shares, with $t<n$ and $1 \leq i \leq n$. Decryption shares are generated by each entity E_i by computing a bilinear pairing. It considers the vote and its private share d_{ID_i} as parameters. Then, the *CE* combines them to decrypt the votes. Finally, the votes are counted and the tally is published.

4.4 Protocol execution

The notation used in our proposed protocol is shown in Table 1.

4.4.1 Voting Set-Up

1) Let $H_1: \{0,1\}^* \longrightarrow \mathbb{G}_1$, $H_2: \mathbb{G}_2 \longrightarrow \{0,1\}^*$ and $H_3: \{0,1\}^* \longrightarrow \{0,1\}^*$ be three hash functions and given two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order prime q satisfying $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. Each E_i sends information to generate its private share as follows:

a) Selects randomly $\boldsymbol{a_{i0} \in Z_q^*}$, keeps it secret and broadcast:

$$a_{i0}P \tag{4}$$

b) Picks up randomly a polynomial $f_i(x)$ over Z_q of at most $t-1$ degree such that $f_i(0) = a_{i0}$.

$$f_i(x) = a_{i0} + a_{i1}x + \cdots + a_{it-1}x^{t-1} \tag{5}$$

Acronym	Meaning
$V =$	Voters
$CE =$	Combining Entity
$E_i =$	All the political parties, some civilians, official and a federal authority registered in the voting process. $1 \leq i \leq n$
$Pu/Pr=$	Public/private key used by the encryption primitive
$t =$	Threshold of the electronic voting protocol
$q_i/s_i =$	i-share of the public/private key assigned to every E_i.
$Q_{ID} =$	Selected entity's public key
$d_{ID_i} =$	i-private share assigned to every E_i
$PBB_s/PBB_s =$	Private/ public key used by signature primitive

Table 1. Notation used in our electronic voting protocol

c) Computes and broadcast $a_{ij}P$ for $j = 1,2,\ldots,t-1$ and sends $f_i(j)$ to each E_j for $j = 1,2,\ldots,n$ where $j \neq i$

2) Once each E_i receives information from other shareholders, they perform the following calculations:

a) After receiving $f_j(i)$ from E_j for $j = 1,2,\ldots,n; j \neq i$, each E_i verifies $f_j(i)P$ by checking:

$$f_j(i)P = \sum_{k=0}^{t-1} i^k a_{jk}P \tag{6}$$

If the check fails, each E_i broadcasts a complaint against E_j.

b) Each E_i computes its private share and keeps it in secret

$$s_i = \sum_{k=1}^{n} f_k(i) \tag{7}$$

Then, it calculates its public share and also keeps it in secret:

$$q_i = s_i P \tag{8}$$

And finally it computes public key:

$$Pu = \sum_{i=1}^{n} a_{i0} P \quad (9)$$

3) Public key is:

$$Pu = PrP \quad (10)$$

And private key, which has been distributed to every entity E_i, is:

$$Pr = \sum_{i=1}^{n} a_{i0} \quad (11)$$

4) Each entity E_i computes its respective private share to encrypt d_{ID_i}, which is associated to its identity *ID,* as follows:

$$d_{ID_i} = s_i Q_{ID} \quad (12)$$

with

$$Q_{ID} = H_1(ID) \in \mathbb{G}_1 \quad (13)$$

5) In order to generate the key pair used to sign the votes, the President of the Ballot Box (PBB) makes the following computations:

a) Private key is PBB_s

$$PBB_s = x \in Z_q^* \quad (14)$$

b) Public key is PBB_p

$$PBB_p = xP \quad (15)$$

4.4.2 Authentication

1) In order to verify if the voter *V* is a valid voter, officials O*,* ask *V* to show its identity card. Then, *O* verify that voter´s name appears on a valid voters list.

2) If the voter's name appears in the list, it is allowed to vote.

4.4.3 Voting

1) The voter *V* chooses a candidate and then the vote v is encrypted by the voter as follows:

a) v is coded as an element of $\mathbb{G}_2$.

b) The voter selects $a \in F_q$ and then, in order to get the result of Equation (13).

c) Considering Q_{ID}, the encrypted vote is given by:

$$< U, W > = < aP, v \oplus H_2(\hat{e}(Pu, Q_{ID})^a) > \quad (16)$$

d) The result of aP is sent entity selected.

2) Given private key x, which was generated during the voting set-up by the PBB, and given the encrypted vote $< U, W > \in \{0,1\}^*$, the voter *V* asks to a blind signature as follows:

a) It chooses randomly $r \in F_q^*$, then it computes $< U, W >'$, by using the Map-to point hash function H_1, given in voting set-up. Then, it is sent to the PBB.

$$< U, W >' = rH_1(U, W) \quad (17)$$

b) PBB computes $\sigma'$and sends it back to the *V.*

$$\sigma' = x < U, W >' \quad (18)$$

c) Then, *V* computes the signature σ as follows:

$$\sigma = r^{-1}\sigma'' \quad (19)$$

3) A store device stores the encrypted vote, the signed and encrypted vote, a time stamp and a hash value, which is gotten by using the signed and encrypted vote and the time stamp.

$$< U, W >, \sigma(< U, W >) \parallel Time\ Stamp \parallel H_3(\sigma(U, W) \parallel Time\ Stamp) \quad (20)$$

4) The voter *V* receives the previously generated hash value as a receipt:

$$H_3(\sigma(U, W) \parallel Time\ Stamp) \quad (21)$$

5) The signed and encrypted vote and its hash value are sent to all entities E_i.

6) The identity card of the voter is invalidated, so it cannot vote more than once.

4.4.4 Counting

1) First, the signature is verified as follows:

a) Given the encrypted vote $< U, W >$ and the signature σ, the Combining Entity *CE* verifies that:

$$\hat{e}\big(PBB_p, H_1(< U, W >)\big) = \hat{e}(P, \sigma) \quad (22)$$

2) Each shareholder E_i calculates its decryption share: $\hat{e}\big(U, d_{ID_i}\big)$. It is sent to the *CE.*

3) The *CE* selects a set $S \subset \{1,2, \ldots n\}$ of t shares $\hat{e}\big(U, d_{ID_i}\big)$ and computes:

$$g = \prod_{i \in S} \hat{e}\big(U, d_{ID_i}\big)^{L_i} \quad (23)$$

Where L_i denotes the appropriate Lagrange coefficient explicitly given by the Equation (2).

4) Once the *CE* calculates g, it recovers plaintext for each vote as follows:

$$v = W \oplus H_2(g) \quad (24)$$

5) All the votes are counted and the tally is published. The voter *V* can check if its vote was counted by verifying if its receipt appears on published tally.

5 Analysis of the protocol

We analyze our protocol from two points of view. The first one details how our protocol meets the security requirements that an electronic voting protocol must meet. The second one involves the performance comparison against protocols have been proposed previously, which use as main construction blocks threshold and signature schemes.

5.1 Security Requirements

There are various electronic voting requirements mentioned in electronic voting. However, we consider those one recommended in [25]. We detail how we meet these requirements as follows:

Privacy: We meet this requirement by using a threshold encryption scheme, which is probably secure against chosen plaintext attack under the computational bilinear Diffie-Hellman problem. With this only the Combining Entity, jointly with at least *t* entities, is the one who is able to decrypt the votes just during the counting stage. The correctness of the aforementioned is proved starting by Equation (23) as follows:

$$
\begin{aligned}
v &= W \oplus H_2(g) \\
W \oplus H_2(g) &= W \oplus H_2\left(\prod_{i\in S} \hat{e}\left(U, d_{ID_i}\right)^{Li}\right) \\
&= W \oplus H_2\left(\hat{e}\left(aP, \sum_{i\in S} L_i s_i Q_{ID}\right)\right) \\
&= W \oplus H_2\left(\hat{e}(aP, PrQ_{ID})\right) \\
&= W \oplus H_2(\hat{e}(aPuPr^{-1}, PrQ_{ID})) \\
&= W \oplus H_2(\hat{e}(aPu, Q_{ID})^a) \\
&= v \oplus H_2(\hat{e}(Pu, Q_{ID})^a) \\
&\qquad \oplus H_2(\hat{e}(Pu, Q_{ID})^a) \\
&= v
\end{aligned}
$$

Eligibility: Only eligible voters participate in the election because they should be registered before the election day and no more than registered voters can cast votes. This requirement is covered during the authentication phase by asking each voter to show its identity card.

Uniqueness: Single one vote per voter will be counted; because the identity card of the voter will be marked in order to such a voter is not able to cast another vote.

Uncoercibility: Any coercers, even authorities, are able to coerce a voter to cast its vote in a particular way. Because the receipt the voter receives, computed from the signed and encrypted vote and a time stamp, does not contain any information which can join the vote with the voter.

Transparency: The hash value of all the votes is published at the end of the voting process to verify, in a transparent way, that all votes were taken into account.

Accuracy: The threshold version of identity based scheme we use, presents the same security properties of the El Gamal cryptosystem, which resists Chosen Plaintext Attack (CPA) considering a decisional Diffie-Hellman assumption over a multiplicative cyclic group. However, it is malleable and does not resist to Adaptive Chosen Ciphertext Attacks (CCA2) [26]. As a consequence, and considering the random oracle model, if the signer acts as a malicious entity, the protocol could be break. In order to prevent such a scenario, we use a hash function and a time stamp. The result of this function is delivered to the voter as a receipt, which assures all cast votes should be counted, and that no one can be altered, deleted, invalidated or copied. Because the voter can check if the hash value that it was delivered appears in the bulletin.

Robustness: We assume that $n \geq 2t - 1$, in such way that at least *t* players are honest. Considering that, each honest entity E_i chooses a random $R \in G1$ and computes w_1 and w_2 as follows:

$$w_1 = \hat{e}(P, R) \in G_2 \tag{25}$$

$$w_2 = \hat{e}(U, R) \in G_2 \tag{26}$$

Then a hash value, denoted by *e,* is calculated:

$$e = hash\left(\hat{e}\left(U, d_{ID_i}\right), \hat{e}(Pu, Q_{ID}), w_1, w_2\right) \tag{27}$$

After that, each entity E_i compute V:

$$V = R + ed_{ID_i} \in G_1 \tag{28}$$

And joins V to the tuple (w_1, w_2, e, V) and then to its share. So, the other entities can check that:

$$\hat{e}(P, V) = \hat{e}(P, R)\hat{e}(q_i, Q_{ID})^e \tag{29}$$

And that:

$$\hat{e}(U, V) = \hat{e}(U, R)\hat{e}\left(U, d_{ID_i}\right)^e \tag{30}$$

The correctness of the aforementioned is proved by checking Equation (29) and Equation (30) as follows:

$$
\begin{aligned}
\hat{e}(P, V) &= \hat{e}(P, R)\hat{e}(q_i, Q_{ID})^e \\
\hat{e}\left(P, R + ed_{ID_i}\right) &= \hat{e}(P, R)\hat{e}(q_i, Q_{ID})^e \\
\hat{e}(P, R)\hat{e}\left(P, ed_{ID_i}\right) &= \hat{e}(P, R)\hat{e}(q_i, Q_{ID})^e \\
\hat{e}(P, R)\hat{e}\left(P, d_{ID_i}\right)^e &= \hat{e}(P, R)\hat{e}\left(s_i P, d_{ID_i} s_i^{-1}\right)^e \\
\hat{e}(P, R)\hat{e}\left(P, d_{ID_i}\right)^e &= \hat{e}(P, R)\hat{e}\left(P, d_{ID_i}\right)^e
\end{aligned}
$$

$$
\begin{aligned}
\hat{e}(U, V) &= \hat{e}(U, R)\hat{e}\left(U, d_{ID_i}\right)^e \\
\hat{e}\left(U, R + ed_{ID_i}\right) &= \hat{e}(U, R)\hat{e}\left(U, d_{ID_i}\right)^e \\
\hat{e}(U, R)\hat{e}\left(U, ed_{ID_i}\right) &= \hat{e}(U, R)\hat{e}\left(U, d_{ID_i}\right)^e \\
\hat{e}(U, R)\hat{e}\left(U, d_{ID_i}\right)^e &= \hat{e}(U, R)\hat{e}\left(U, d_{ID_i}\right)^e
\end{aligned}
$$

Oper.	Ohkubo *et al*	Cramer. *et al*	Baudron. *et al*	Mu. *et al*	Our protocol
Am	3	1	$2(i\text{-}2) + 2$	1	0
M	$16 + t\text{-}1$	$12 + i\text{-}1$	$L*10 + 8 + 2(t\text{-}1)$	6	0
E	13	$19 + n + n*i$	$L\,(n! + 13) + 9 + t$	11	0
I	2	3	$L*2$	1	1
A	NA	NA	NA	NA	0
S	NA	NA	NA	NA	$2 + 1*v + 3*i$
Hash	5	NA	NA	NA	$1*v + 2$
$\hat{e}$	0	0	0	0	$1*v + 1*i + 2$

Table 2. Cryptographic operations developed in our protocol.

Parameters	Ohkubo *et al*	Cramer. *et al*	Baudron. *et al*	Mu. *et al*	Our protocol
Key Pairs	1 * V + 1 * C. Auth + 2	1	1 * V	1 * V+1	2
C.A	1 * V	0	1 * V	1 * V	0
T.A	0	0	0	0	1
Complexity Assumptions	Diffie-Hellman and Product of two prime numbers	Diffie-Hellman	Composite Residuosity Class	Diffie-Hellman	Bilinear Diffie-Hellman

Table 3. Key Pairs and required authorities in our protocol

With this, we assure our protocol can be developed even if there are entities E_i who do not give Combining Entity their decryption shares.

5.2 Performance comparison

In order to compare our protocol with previous work, Table 2 shows the comparison of computation for our electronic voting protocol against other ones based on threshold and blind signature schemes.

The computation depends on the number of cryptographic operations used by each protocol. They are described according to the following notation: Am, M, E and I stands for modular addition, multiplication, exponentiation and inversion respectively. A and S denote addition and scalar multiplication on an elliptic curve. N/A is for Not Available. Moreover, parameter n represents total number of shareholders who participate during the voting process with $1 \leq i \leq n$, t denotes the threshold that the voting protocol considers for counting stage, and v denotes the voters who participate during the voting process. L stands for organization of the authorities, national, regional or local.

All the protocols we considered to make the analysis are based on finite group operations. However, our protocol involves operations within finite fields, as well as field extensions.

In our case, we consider prime finite fields $\mathbb{F}_p$. The embedding degree for that field is $k = 12$ which involves operations on the field extension $\mathbb{F}_{p^{12}}$.

However, in order to get an element of $\mathbb{F}_{p^{12}}$ from $\mathbb{F}_p$, it is necessary to use intermediate field extensions such as: $\mathbb{F}_{p^2}, \mathbb{F}_{p^6}$.

Working with different fields involves the use of polynomials to represents field elements. This technique is known as tower fields, which is used to get a bilinear pairing.

Besides our protocol does not involve the cryptographic operations Am, M and I, it does use several evaluations of bilinear pairings, which involves additions and multiplications over the finite field $\mathbb{F}_p$ and its

extensions. Moreover, it is easy to observe that the cost in our protocol is the highest.

However, we consider high cost operations can be addressed by using a special device [27], which efficiently develops this sort of cryptographic operations. The inclusion of such a device is considered to be cheaper, and then preferred, than a Public Key Infrastructure. Moreover the security degree that our protocol offers is better than previous protocols. It is because the security of our protocol relies on the hardness of the computational Diffie-Hellman problem (CDH) and the bilinear Diffie-Hellman problem (BDH). And so far, there does not exist any algorithm that solve BDH problem in a polynomial time.

According to the aforementioned we stand that our protocol is a good improvement to the currently existing electronic voting protocols based on Threshold Cryptography, mainly because its security features.

Table 3 shows a comparison in terms of the total number of keys pairs and required authorities by our electronic voting protocol and other ones. In it, L is the number of levels and V is the number of voters that the protocol considers. C.A and T.A mean Certification and Trust Authority respectively.

It is possible to see that in previous protocols the number of key pairs and C.A's increase depending on the number of voters. Moreover, even Cramer's protocol do not need a T.A, the complexity assumptions we use, BDH Problem for instance, are stronger than those one used in such a protocol, becoming our protocol more secure than previous protocols.

6 Conclusions and Future work

We present a protocol based on two cryptographic primitives from bilinear pairings. The first one is a threshold scheme without a Private Key Generator and the second one is blind signature scheme. It meets the following electronic voting security requirements: privacy, eligibility, uniqueness, uncoercibility, transparency, accuracy and robustness. Three of them, privacy, accuracy and robustness are addressed by using bilinear pairings

This protocol shows the main reasons of changing the use of Public Key Cryptography by Identity based Cryptography, as a future work we propose to change de signature primitive by another one which is based on identity, becoming an electronic voting protocol from Identity based Public Key Cryptography.

Another future work we consider the use of multisignature schemes. These schemes allow any subgroup of users to sign a document jointly, so that a verifier is convinced that each member of the subgroup participate in the signing process. We also consider incorporating threshold blind signatures in our protocol, so the private key will be distributed among *n* parties. In this kind of protocols a vote is signed and any subset of more than *t* parties is able to use their shares and obtain a blind signature, which can be verified by anybody using the unique fixed public key. Moreover we will continue our research over the distributed responsibility idea.

Finally, as part of the design of new cryptographic protocols, we will test the security of our protocol under formal security.

References:

[1] H. M. El-Bakry and N. Mastorakis, "A Real-Time Intrusion Detection Algorithm for Network Security", *WSEAS Transactions on Communications,* Issue 12, Volume 7, pp: 1222 – 1234, December 2008.

[2] D. L. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms", *Communications of the ACM*, Vol.24, No.2, pp.84-88 February 1981.

[3] D. L. Chaum, "Elections with Unconditionally-Secret Ballots and Disruption Equivalent to Breaking RSA", in *Advances in Cryptology EUROCRYPT '88*, LNCS 330, Springer Verlag, Berlin, pp.177-182, 1988.

[4] R. Cramer, R. Gennaro, and B. Schoenmakers, "A Secure and Optimally Efficient Multi-Authority Election Scheme", in *Advances in Cryptology EUROCRYPT '97*, LNCS 1233, Springer Verlag, Berlin, pp.103-118, 1997.

[5] NIST SP 800-32, Introduction to Public Key Technology and the Federal PKI Infrastructure, 2001

[6] A. Shamir, "Identity-based cryptosystems and signature schemes", in *Advances in Cryptology-Crypto´84*, LNCS 196, Springer-Verlag, pp. 47-53, 1985.

[7] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing", in *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology,* LNCS 2139, Vol. 2139, pp. 213-229, 2001.

[8] M. Salinas Rosales, G. Gallegos Garcia and G. Duchen Sanchez, "Efficient Message Authentication Protocol for WSN", *WSEAS Transactions on Computers,* Issue 6, Volume 8, pp: 895 – 904, June 2009.

[9] W.Diffie, M. Hellman. New directions in cryptography. *In IEEE Transaction on Information Theory*, Vol. 22, No.6, pp.644-654,1976.

[10] C. Cocks, An Identity Based Encryption Scheme Based on Quadratic Residues, Cryptography and Coding - Institute of Mathematics and Its Applications *International Conference on Cryptography and Coding, In Proceedings of IMA 2001*, LNCS 2260, pages 360-363, Springer-Verlag, 2001.

[11] D. Chaum. Blind Signatures for untraceable payments. *Advances in Cryptology- Crypto82*, LNCS, eds. Plenum D. Chaum, R. L. Rivest and A. T. Sherman. Plenum Press, New York, pp. 199-203, 1983.

[12] A. Boldyreva, "Efficient Threshold Siganture, Multisignature and Blind Signature Schemes based on the Gap-Diffie-Hellman-Group Siganture Scheme", in *Proceedings of International Workshop on Public Key Cryptography 2003, PKC 2003,* LNCS 2139, pp. 31-46, Springer-Verlag, 2003.

[13] F. Zhang, K. Kim. ID-Based Blind Siganture and Ring Signature from Pairings. Advances *in Cryptology-Asyacrypt 2002*, LNCS 2510, Springer-Verlag, 2002.

[14] Y. Desmedt. Threshold Cryptosystems. *Advances in Cryptology-ASIACRYPT92*, Old Coast, Queensland, December, LNCS 718, eds. J. Seberry and Y Zheng. Springer-Verlag, Berlin, 3-14 (Invited paper), (1993).

[15] A. Shamir. How to share a secret. Communications *ACM 22*, pp. 612-613, 1979.

[16] J. Baek, Y. Zhen. Identity-Based Threshold Decryption. *Cryptology ePrint Archive*, Report 2003/164, available at http://eprint.iacr.org/2003/164, PCK 2004, 2004.

[17] B. Libert and J. Quisquarter, "Efficient Revocation and Threshold Pairing Based Cryptosystems", *Symposium on Principles of Distributed Computing,* PODC 2003, pp. 163-171, 2003

[18] P. Barreto, H. Kim and M. Scott, "Efficient algorithms for pairing-based cryptosystems", in *Advances in Cryptology-Crypto'02*, LNCS 2442, Springer-Verlag, pp. 354-368, 2003.

[19] M. Ohkubo, F. Miura, M. Abe, A. Fujioka and T. Okamoto, "An improvement on a practical secret voting scheme", in *Proceddings of the Second International Workshop on Information Security'99*, pp. 225–234, 1999.

[20] O. Baudron, P. Fouque, D. Pointcheval, G. Poupard and J. Stern, "Practical multi-candidate election system", In *PODC '01*, pp 274–283, ACM, 2001

[21] P. Paillier, "Public-Key Cryptosystems Based on Discrete Logarithms Residues", in *Proceedings of Eurocrypt '99*, LNCS 1592. Springer-Verlag, 1999.

[22] Y. Mu and V. Varadharajan, "Anonymuos secure e-voting over a network", in *Proceedings of the 14th Annual Computer Security Applications Conference, IEEE Computer Society*, pp. 293 – 299, 1998.

[23] G. Gallegos-G, R. Gomez-C, M. Salinas-R and G.I. Duchen-S, "A New and Secure Electronic Voting Protocol Based on Bilinear Pairings", in *Proceedings of the IEEE International Conference on Electrical, Communication and Computers CONIELECOMP 2009, IEEE Computer Society,* pp 240 – 244, 2009.

[24] D. Liem, F. Zhang and K. Kim, "A New Threshold Blind Signature Scheme from Pairings", *The 2003 Symposium on Cryptography and Information Security*, Hamamatsu, Japan, 2003

[25] O. Cetinkaya and D. Cetinkaya, "Verification and Validation Issues in Electronic Voting" *The Electronic Journal of e-Government* Vol. 5, Issue 2, pp 117 - 126, available online at www.ejeg.com, 2007

[26] D. Pointcheval, "Fundamental problems in provable security and cryptography", *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences,* Vol. 364, Issue 1849, pp. 3215-3230, 2006.

[27] K. A. Gwalani and O. Elkeelany, "Design and Evaluation of FPGA Based Hardware Accelerator for Elliptic Curve Cryptography Scalar Multiplication". *WSEAS Transactions on Computers,* Issue 5, Volume 8, pp: 884 – 893, May 2009.