Identity-Based Threshold Cryptography for Electronic Voting

GINA GALLEGOS-GARCÍA[1], ROBERTO GÓMEZ-CÁRDENAS[2], GONZALO I. DUCHÉN-SÁNCHEZ[1]
[1]Graduate School, [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
gina@calmecac.esimecu.ipn.mx

Abstract: - 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. Different algorithms, based on public key schemes, have been proposed in the literature to meet these security requirements. We propose the use of threshold cryptography and 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. We make a comparative analysis of our proposal with other electronic voting protocols. It is based in their performance and the cryptographic primitives they use.

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

1 Introduction

Since 1964, electronic voting has been mentioned in different media as the use of computers or computerized voting equipment to cast ballots in an election. It has been mentioned as a reasonable alternative to conventional elections and other opinion expressing processes. However, it must offer the same benefits as a conventional election does.

An electronic voting protocol involves tree 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 can vote. The tallying authorities collect the cast votes and tally the results of the election. All these actors 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, the 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.

Electronic voting protocols in the literature can be classified into three basic types: protocols based on mix-nets [1], protocols based on blind signatures [2] and protocols based on threshold cryptography [3]. All of them are based on asymmetric or public key cryptography PKC, which offers high flexibility through key agreement protocols and authentication mechanisms. However, it is necessary to implement a public key infrastructure PKI [4] to provide certificates which bind public keys to entities, and enable other ones to verify public key bindings. As a consequence, the components of the protocol increase notably.

An alternative to PKI is the identity based cryptography (IBC). With this kind of cryptography, it is possible to have all the benefits offered by PCK, but without the need of certificates and a PKI infrastructure.

IBC was proposed by Shamir in 1984 [5], but the first implementation was made by Boneh in 2001[6]. An identity based cryptosystem is a cryptosystem in which the public key is retrieved from an identity of the entity, and the private key is securely distributed by a Key Distribution Center. Most common IBC implementations are based on bilinear pairings.

We propose the use of threshold cryptography from bilinear pairings into electronic voting protocols, to ensure security requirements without the use of certificates neither a PKI infrastructure. In order to compare our work with previous proposals, we present a comparative analysis of electronic voting protocols.

The remainder of this document is organized as follows. In section 2 related work is presented. Section 3 introduces the new electronic voting protocol. Section 4

introduces the results obtained with our protocol, by comparing it with those based on threshold cryptography. Finally section 5 presents our conclusions and draft further work for this research.

2 Related Work

In [7], 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 [3] 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* [8] 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 [9] 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 [10] 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.

The four previous protocols are based in public key cryptography. They need a public key infrastructure PKI to manage all the certificates needed to verify the public key owner's identity. The cost and complexity of the PKI infrastructure makes it difficult to integrate in electronic vote protocols. Identity based cryptography was created as a means of overcoming this problem.

In [11], 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. However this protocol has the disadvantage that the 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.

In order to solve this issue, we propose to eliminate the PKG. The entities described in Section 1 will create the public and private keys used to encrypt and decrypt the votes.

3 Protocol Description

3.1 Considerations

The following considerations have been adopted as a part of the environment under the protocol will operate:

- 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, $E(K): y^2 + ay = x^3 + bx^2 + cx + d$, 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.

3.2 Complexity Assumptions

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

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 [6, 13].

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.
- Gap Diffie-Hellman Problem (GDHP): A class of problems where the CDH problem is hard but DDH problem is easy.

Groups where the CDH problem is hard but the DDH problem is easy are called Gap Diffie-Hellman (GDH) groups.

3.3 Our electronic voting protocol

Our proposal is based on two cryptographic primitives, the threshold version of the Boneh-Franklin identity based encryption scheme [14] and the blind signature scheme [15]. In [14] 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 [12] 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.

3.3.1 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 a private key Pr at the counting phase. The generation of this key pair involves the participation of n entities, 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. The public share is used to generate the public key Pu used by the voters, to encrypt the votes during the voting stage. The private share is used during the counting stage to generate a private key Pr in order to decrypt the votes. With the intention of guarantee anyone be able to send false information, the private and the public shares must be kept in secret.

Another key pair generated in this stage is the private/public key pair of the President of Ballot Box, 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.

3.3.2 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 ofiicials and all the voters.

3.3.3 Voting

A candidate is selected by the voter and its vote is encrypted with the public key Pu. Then, it is blindly signed with the 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.

3.3.4 Counting

In this stage, the votes are verified, decrypted and counted by the Combining Entity, *CE*, is in charge of verifying the signature and decrypting the votes. The signatures of the votes are verified with the public key of the President of the Ballot Box, PBB_p, and with the intention of decrypt the votes, the *CE* selects t of n decryption shares, with $t<n$ and $1 \leq i \leq n$. The decryption shares are generated by each entity E_i by computing a bilinear pairing. It considers the vote and the private share s_i as parameters. Then, the *CE* combines them to decrypt the votes. Finally, the votes are counted and the tally is published.

3.4 Protocol execution

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

Acronym	Meaning
$PBB =$	President of Box Ballot
$V =$	Voters
$O =$	Officials
$CE =$	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 of the encryption primitive
$t =$	Threshold of the Encryption Scheme
$q_i/s_i =$	i-share of the public/private key assigned to every E_i.

Table 1. Notation used in our electronic voting protocol

3.4.1 Voting Set-Up

1) Given two groups $\mathbb{G}_1, \mathbb{G}_2$ 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 $a_{i0} \in Z_q^*$, keeps it secret and broadcast:

$$a_{i0}P \qquad (1)$$

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

$$f_i(x) = a_{i0} + a_{i1}x + \cdots + a_{it-1}x^{t-1} \qquad (2)$$

c) Computes and broadcast $a_{ij}P$ for $j = 1,2,\dots,t-1$ and sends $f_i(j)$ to each E_j for $j = 1,2,\dots,n$ 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 $j = 1,2,\dots,n; j \neq i$, each E_i verifies $f_j(i)P$ by checking:

$$f_j(i)P = \sum_{k=0}^{t-1} i^k a_{jk}P \quad (3)$$

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

$$s_i = \sum_{k=1}^{n} f_k(i) \quad (4)$$

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

$$q_i = s_i P \quad (5)$$

And finally it computes the public key:

$$Pu = \sum_{i=1}^{n} a_{i0}P \quad (6)$$

3) The public key is:

$$Pu = PrP \quad (7)$$

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

$$Pr = \sum_{i=1}^{n} a_{i0} \quad (8)$$

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

a) Let $H: \{0,1\}^* \rightarrow \mathbb{G}_1$ be a Map-to point hash function.

b) The private key is $PBB_s = x \in Z_q^*$ (9)

c) The public key is $PBB_p = xP$ (10)

3.4.2 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.

3.4.3 Voting

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

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

b) The encrypted vote is given by:

$$< U, W > = < rP, v \oplus H_2(\hat{e}(P,Q)^r) > \quad (11)$$

2) Given the 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 $r \in Z_q^*$, then it computes $< U, W >'$, by using the Map-to point hash function *H,* given in voting set-up. Then, it is sent to the PBB.

$$< U, W >' = rH(U, W) \quad (12)$$

b) PBB computes σ' and sends it back to the *V*.

$$\sigma' = x < U, W >' \quad (13)$$

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

$$\sigma = r^{-1}\sigma'' \quad (14)$$

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.

$$< U, W >, \sigma(< U, W >) \parallel Time\ Stamp \parallel Hash\ (\sigma(U, W) \parallel Time\ Stamp) \quad (15)$$

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

$$Hash(\sigma(U, W) \parallel Time\ Stamp) \quad (16)$$

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.

3.4.4 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:

$$\hat{e}\left(PBB_p, H(< U, W >)\right) = \hat{e}(P, \sigma) \quad (17)$$

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

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

$$g = \prod_{i \in S} \hat{e}(U, s_i P)^{L_i} \quad (18)$$

Where L_i denotes the appropriate Lagrange coefficient explicitly given by the formula:

$$L_i = \prod_{i \in S, j \neq i} \frac{-x_j}{(x_i - x_j)} \quad (19)$$

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

$$v = W \oplus H_2(g) \quad (20)$$

4) 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 the published tally.

4 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.

4.1 Security Requirements

There are various e-voting requirements mentioned in electronic voting protocols. However, we consider those one recommended in [16]. We detail how we meet these requirements as follows:

Privacy: We meet this requirement by using a blind signature scheme, which is secure against one more forgery under chosen message attack assuming the hardness of chosen-target CDH problem [12]. As a result of this, there is no way to associate a voter with a vote.

Eligibility: Only eligible voters participate in the election because they should be registered before the election day and only registered eligible voters can cast votes. This requirement is covered during the authentication phase by asking each voter to show its identity card.

Uniqueness: Only 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) [17]. 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 any vote cannot be altered, deleted, invalidated or copied.

4.2 Performance comparison

In order to compare our protocol with previous work, we adopt the evaluation used in [12]. Table 2 shows the comparison of computation for our electronic voting protocol against other ones based on threshold cryptography.

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.

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. Performance comparison in our protocol

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: $\mathbb{F}_{p^2}, \mathbb{F}_{p^6}$.

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, high cost operations can be addressed by using a special device, 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.

5 Conclusions and Future work

We present a protocol based in identity-based threshold cryptography without a private key generator, which use two cryptographic primitives from bilinear pairings. It meets the following electronic voting security requirements: privacy, eligibility, uniqueness, uncoercibility, transparency and accuracy. Our protocol is efficient by considering use a special device, which computes all the cryptographic operations and it does not require a PKI infrastructure.

This protocol shows the main reasons of changing the use of publik key criptography 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 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.

References:

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

[2] D. Chaum, "Elections with Unconditionally-Secret Ballots and Disruption Equivalent to Breaking RSA", in *Advances in Cryptology EUROCRYPT '88*, Lecture Notes in Computer Science 330, Springer Verlag, Berlin, pp.177-182 (1988).

[3] R. Cramer, R. Gennaro, and B. Schoenmakers, "A Secure and Optimally Efficient Multi-Authority Election Scheme", in *Advances in Cryptology EUROCRYPT '97*, Lecture Notes in Computer Science 1233, Springer Verlag, Berlin, pp.103-118 (1997).

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

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

[6] D. Boneh, 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.

[7] 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.

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

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

[10] 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.

[11] G. Gallegos-G, R. Gomez-C, M. Salinas-R, G.I. Duchen-S, "A New and Secure Electronic Voting Protocol Based on Bilinear Pairings", *IEEE International Conference on Electrical, Communications, and Computers CONIELECOMP 2009,* pp 240 – 244, Puebla-Mexico, 2009.

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

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

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

[15] 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.

[16] 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

[17] 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.