Revista Facultad de Ingeniería Universidad de Antioquia
ISSN: 0120-6230
revista.ingenieria@udea.edu.co
Universidad de Antioquia
Colombia

López García, Lourdes; López Chau, Asdrúbal; Silva Pérez, Javier; León Chávez, Miguel
An e-voting system for Android Smartphones
Revista Facultad de Ingeniería Universidad de Antioquia, núm. 72, septiembre-, 2014, pp. 9-19
Universidad de Antioquia
Medellín, Colombia

Available in: http://www.redalyc.org/articulo.oa?id=43031750002

How to cite
Complete issue
More information about this article
Journal's homepage in redalyc.org

Scientific Information System
Network of Scientific Journals from Latin America, the Caribbean, Spain and Portugal
Non-profit academic project, developed under the open access initiative

An e-voting system for Android Smartphones

Sistema de votación electrónica para teléfonos inteligentes Android

Lourdes López García[1], Asdrúbal López Chau[1], Javier Silva Pérez[2], Miguel León Chávez[3]*

[1]Centro Universitario UAEM Zumpango, Computer Engineering Department, Camino viejo a Jilotzingo continuación calle Rayón, Valle Hermoso. C.P. 55600. Estado de México, México.

[2]Centro de Investigación y de Estudios Avanzados del IPN, Computer Science Department, Av. IPN 2508, San Pedro Zacatenco. C.P. 07360. México, D.F., México.

[3]Benemérita Universidad Autónoma de Puebla, Computer Science Department. Av. San Claudio y 14 sur, Ciudad Universitaria. C.P. 72570. Puebla, México.

(Received February 15, 2013; accepted May 06, 2014)

Abstract

Electoral processes using electronic devices allow voters to cast electronically. Devices such as personal computers, Direct Record Machines (DRE) voting machines, and smartcards, among others, in conjunction with private nets or the Internet have been the main tools to implement electronic voting systems (e-voting). Some years ago, mobile devices had not been considered due to their storage restrictions and low computational power; however, nowadays the technology in these devices has advanced and it is possible to implement complicated applications that offer security services such as confidentiality and authentication. In this paper, a reliable and efficient e-voting system for Android Smartphone is implemented. The e-voting proposed herein is composed of three entities: the authentication server (AS), the voting server (VS) and the voter (V) which runs on an Android Smartphone. Two short messages, between V and AS, are necessary to generate an electronic ballot. In order to cast a vote, one more message containing the ballot is sent from V to VS. Bilinear pairing functions are used to verify the signatures contained in the electronic ballot. Each ballot has short lengths, implying improvements in transmission times. Additionally, considering that the most costly operations such as bilinear pairing are not performed in V, the results obtained in tests show that the e-voting system is efficient.

* Corresponding author: Lourdes López García, e-mail: mllopezg@uaemex.mx; phone number: + 52 591 9174140, (L. López)

----------*Keywords:* E-voting, digital signatures, bilinear pairing, mobile devices

Resumen

Los procesos electorales que utilizan dispositivos electrónicos permiten a los votantes emitir su voto de forma electrónica. Dispositivos tales como las computadoras de escritorio, las máquinas de registro directo (DRE), tarjetas inteligentes, etc., junto con las redes privadas o el Internet han sido las herramientas principales para implementar sistemas electrónicos de votación. Años atrás, los dispositivos móviles no habían sido considerados debido a sus restricciones en almacenamiento y poco poder de procesamiento; sin embargo, en la actualidad, la tecnología en estos dispositivos ha evolucionado, y es posible implementar aplicaciones cada vez más completas, que consideran los servicios de seguridad como son la confidencialidad y la autenticación. En este artículo, un sistema de votación electrónica (e-voting) es implementado para teléfonos inteligentes Android. El sistema está compuesto de tres entidades: Servidor de Autenticación (SA), Servidor de Votación (SV) y el votante (V) el cual se ejecuta en un teléfono inteligente Android. Dos mensajes de longitud corta son necesarios para generar una boleta electrónica entre el V y el SA. Con la finalidad de emitir el voto, la boleta es enviada en un mensaje más, del V al SV. Las funciones de emparejamiento bilineales son usadas para verificar las firmas contenidas en la boleta electrónica, la cual tiene una longitud corta, por tanto, el tiempo de transmisión es muy corto. Adicionalmente, considerando que las operaciones más costosas, como el emparejamiento bilineal, no son ejecutadas en el V, los resultados obtenidos de las pruebas muestran que el sistema de votación electrónica es muy eficiente.

----------*Palabras clave:* Votaciones electrónicas, firmas digitales, emparejamientos bilineales, dispositivos móviles

Introduction

Until few years ago, electronic devices such as desktop computers, DRE voting machines, smart cards, etc. along with private nets or the Internet had been the key components of most e-voting systems. Recently, the advances in technology have allowed sophisticated implementations for different mobile devices such as Tablets and Smartphones. Nowadays, all Smartphones have access to the Internet, and a large number of electronic transactions occur at every moment. Currently, it is possible to implement applications with hard constraints that include the use of security services such as the confidentiality and the authentication.

As is known, an e-voting is a prime example in which the use of security services is quite necessary in order to cast a vote in a private way; in addition, it is necessary to consider the requirements that a secure e-voting must meet [1].

- Voter privacy (Anonymity): any entity must not be able to associate a voter with a vote. This requirement must be preserved during and after the election.
- Eligibility: only eligible voters participate in the election.
- Uniqueness: just one vote per voter should be counted.

- Fairness: no partial tally is revealed before the end of the voting period, i.e., electoral authorities should not be able to anticipate the results before the tally.
- No-coercion: any entity should not be able to extract the value of the vote from the ballot.
- Accuracy: tally should be correctly computed from correctly cast votes.
- Receipt-freeness: the system should not provide a confirmation of the receipt of the vote, which can be used to prove the value of the vote to a third party.

To fulfill the majority of previously listed requirements, many secure e-voting protocols have been proposed by several authors. In literature, two main classes can be found, mainly, protocols based on blind signatures and protocols based on homomorphic functions [2].

Among the e-voting systems implemented for mobile devices, it can be mentioned several systems that use different applications and cryptography operations. For instance, in [3] is proposed an e-voting scheme based on VoteBox system which employs homomorphic encryption of ElGamal variant, non-interactive Zero Knowledge and a bulletin board. Another system is proposed in [4] which is a voting software that can be started from within an application called E-Chalk; it is an e-voting in the classroom for educational proposes. In [5] is developed an efficient mobile voting system based on elliptic curves. They used a combination of the elliptic curve cryptography with the ElGamal encryption and the symmetric cryptography with the AES128-bit to obtain better efficiency. Finally, [6] proposed an e-voting scheme which is the version to mobile devices of the Caltech/Mit Voting System. They used the distributed Paillier's encryption scheme as the main cryptography operation. None of which have considered implementations using bilinear pairing cryptography.

In this work, an e-voting system suitable for mobile devices (Smartphones) with Android Operating System (O.S.) is introduced. The e-voting system offers a reliable election through a secure and efficient e-voting protocol. The system is based on the e-voting protocol proposed in [7] to generate secure small ballots. These ballots consist of two bilinear pairing digital signatures with their respective messages. In the revised literature, there are not e-voting systems similar to the presented in this paper. Other implementations use different security schemes and are developed for other operating systems. The use of Android operating system is supported on the fact that it is the most popular operating system for Smartphones, currently reaching more than 50% of the smartphone market.

The rest of this paper is organized as follows. Section II describes the mathematical background used by the cryptography tools that are utilized by the system. Section III presents the architecture of the e-voting system. Section IV shows the implementation of a cryptography library. Section V presents the functionality of the e-voting scheme among the Voter and the Servers. Finally, conclusions are given in Section VI.

Mathematical background

The proposed e-voting system is based on bilinear pairing according to the layer model showed in figure 1. The first layer is considered the modular arithmetic of prime field F_P. The second layer consists of both basic operations of elliptic curves (addition and double of points, scalar multiplication) and the basic operations of the tower field constructed by F_P, F_{Pk1}, F_{Pk1k2}, which are used by the bilinear pairing function. The third layer has both the bilinear pairing function and the special function called mat-to-point which are the main blocks in schemes based on bilinear pairings presented in the fourth layer. The next layer consists of the implementation of security protocols among the electoral entities and the voter. Finally, the sixth layer is considered the e-voting system on Smartphone.

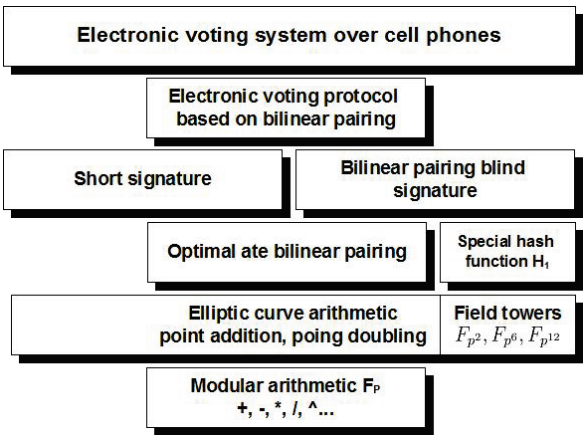


Figure 1 Layer model of e-voting system based on pairings

A. Elliptic curves and bilinear pairings

Let $E(F_{pm})$ be an elliptic curve defined over finite field F_{pm} as in [8]. Given the elliptic points P, Q, R, $S \in E(F_{pm})$, the operations $R=P+Q$ and $S=P+P$ are defined as point addition and point doubling respectively. Let k be a positive integer and $P \in E(F_{pm})$. Then the elliptic curve scalar multiplication is the operation that computes the multiple $Q=kP$, defined as the point resulting of adding $P+ \cdots + P$, k times.

Let r be a prime number. Let (G_1, G_2) be two additive groups with identity O and G_T be a multiplicative group with 1 as its identity, all of them of order r. The bilinear pairing is defined on (G_1, G_2) as a function $\hat{e}:G_2 \times G_1 \rightarrow G_T$ satisfying the following properties:

- Bilinearity: for all $P,P' \in G_1$, Q, $Q' \in G_2$
- $\hat{e}(Q+Q',P)= \hat{e}(Q,P) \cdot \hat{e}(Q',P)$ and $\hat{e} = (Q,P+P) = \hat{e}(Q,P) \cdot \hat{e}(Q,P')$
- Non-degeneracy: $\hat{e}(Q,P)=1$ only if $P=$O or $Q=$O
- $\hat{e}$ can be efficiently computed.

The bilinearity property implies that, given $a, b \in Z_r$

$$\hat{e}(aQ,bP)=\hat{e}(bQ,aP)=\hat{e}(Q,abP)=\hat{e}(abQ,P)=\hat{e}(Q,P)^{ab}$$

When G_1, G_2, the bilinear pairing is called symmetric, otherwise is named asymmetric.

B. Special map to point function

H_1 is a special hash function called *map-to-point*. This function maps an arbitrary string into a point in an elliptic curve [9]. Let G_1 be an additive group, the special function map-to-point is defined as $H_1 :\{0,1\}^* \rightarrow G_1$. To guarantee the security of the function H_1 must fulfill all properties of the standard hash functions.

C. Short signature scheme

The short signature proposed in [10], is a useful digital signature in environments with bandwidth constrains. This scheme has the property that the Computational Diffie-Hellman Problem is hard, but the Decisional Diffie-Hellman Problem is easy on certain additive groups generated by base points on an elliptic curve defined over a finite field.

This short signature scheme consists of three algorithms defined as follows: let $G_1=\langle P \rangle, G_2= \langle Q \rangle$ be additive groups of order prime r where P and Q are points over an elliptic curve.

1. *Key generation.* Pick a random integer $d \in Z_r$, and compute $V=dQ$. The public key is $V \in G_2$ and the private key is d.
2. *Signing.* Given a private key d, a message $m \in \{0,1\}^*$, compute $S=dH_1(m) \in G_1$
3. *Verification.* Given a public key V, a message m, and a signature S, verify that $(Q,V,H_1(m),S)$ is a valid Diffie-Hellman tuple, i.e., if DDHP is solved, the signature is valid, otherwise it is invalid. Using the bilinear pairing function, the verification checks whether $\hat{e}(Q,S)=\hat{e}(V,H_1(m))$ holds.

D. Blind signature scheme

A blind signature is a class of digital signature where the message is *hidden* before signed [11]. The process to generate a blind signature requires two additional algorithms only known by the

signature requester: blinding c and unblinding c' such that $c'(s(c(m)))=s(m)$, where s is the signing function and m is the message.

The blind signature scheme proposed in [12] consists of five algorithms and uses the domain parameters defined as the same way that the short signature of Boneh.

1. *Key generation.* Pick a random integer $d \in Z_r$, and compute $V=dQ$. The public key is $V \in G_2$ and d is the private key.
2. *Blinding* (user). Given a message m, calculate $M=H_1(m)$, randomly find $b \in Z_r^*$ and compute $\hat{M}=bM$.
3. *Signature* (signer). Given a blind message $\hat{M}$ and private key of the signer d, compute $\hat{S} = \hat{M}$.
4. *Unblinding* (user). Given a blind signature $\hat{S}$ and a blind factor b, calculate $S=b^{-1}\hat{S}$; Then $S \in G_1$ is the signature of the message m.
5. *Verification* (third party). Given a message m, a signature S, and the public key V of the signer, check whether $\hat{e}(Q,S)=\hat{e}(V,H_1(m))$ holds.

Architecture of the e-voting system

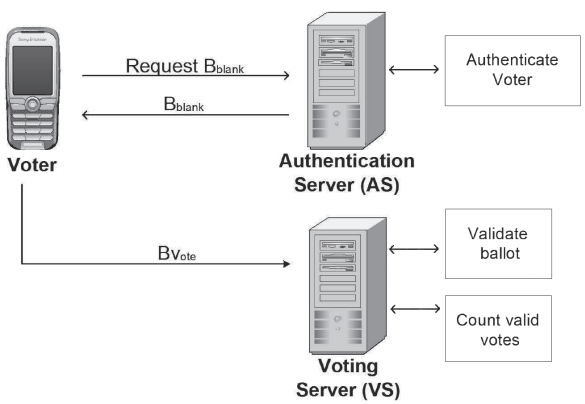


Figure 2 Architecture of the proposed e-voting system over Smartphone

The architecture of the proposed e-voting system is presented in figure 2. It is composed of two servers and one or more Android Smartphones. The e-voting protocol has two electoral entities and the voter. The interaction between the voter and these electoral entities is divided into four phases:

***Registration phase**:* the voter must be registered by an electoral entity called Register Server (RS) which generates the nominal list that contains all valid voters and their electoral identities.

***Authentication phase**:* the goal during this phase is to generate a blank ballot for all valid voters registered in the nominal list. The blank ballot, signed by Authentication Server (AS), consists of one digital signature which will certificate the validity of the ballot in the voting phase.

***Voting phase**:* in this phase, the voter generates and sends the ballot to Voting Server (VS), which is responsible to receive, verify and storage it. The ballot consists of both the authorization signature generated by AS and the signature of the vote to protect it of possible modifications. To receive the ballot, the VS verify both signatures. If both signatures are valid, then the ballot is stored as valid, otherwise it is invalid. Finally, the VS generates an acknowledge using all information of the ballot and sends it to the voter.

***Counting phase**:* the VS tell the votes and publish the result of the election.

E-voting protocol dataflow

In a previous phase called *Registration phase*, the public keys of the voters as well as the electoral servers are generated by an RS. The RS sends the nominal list to the AS. This list contains all information of the voters, including their digital certificates. Finally, the electoral servers publish their public keys for verifying the signatures.

The steps to produce an electronic ballot are just the generation of two digital signatures (short signature of Boneh and blind signature of Boldyreva mentioned above) which require the minimal cryptography operations. The protocol

between the voter and the electoral entities [7], presented in figure 3, uses as domain parameters the tuple (G_1, G_2, r, P, Q, H_1) and the next notation:

- ID_X, d_X, V_X: Identifier and private/public key pair of the entity X.
- d_t, V_t: Alias private key and alias public key generated by the Voter.
- b: Denotes a random blind factor generated by the Voter.
- $m2s$: $G_2 \rightarrow \{0.1\}^k$: Is the function that maps a point in the additive group G_2 to a string of k-bits length.
- H: Denotes a hash function.

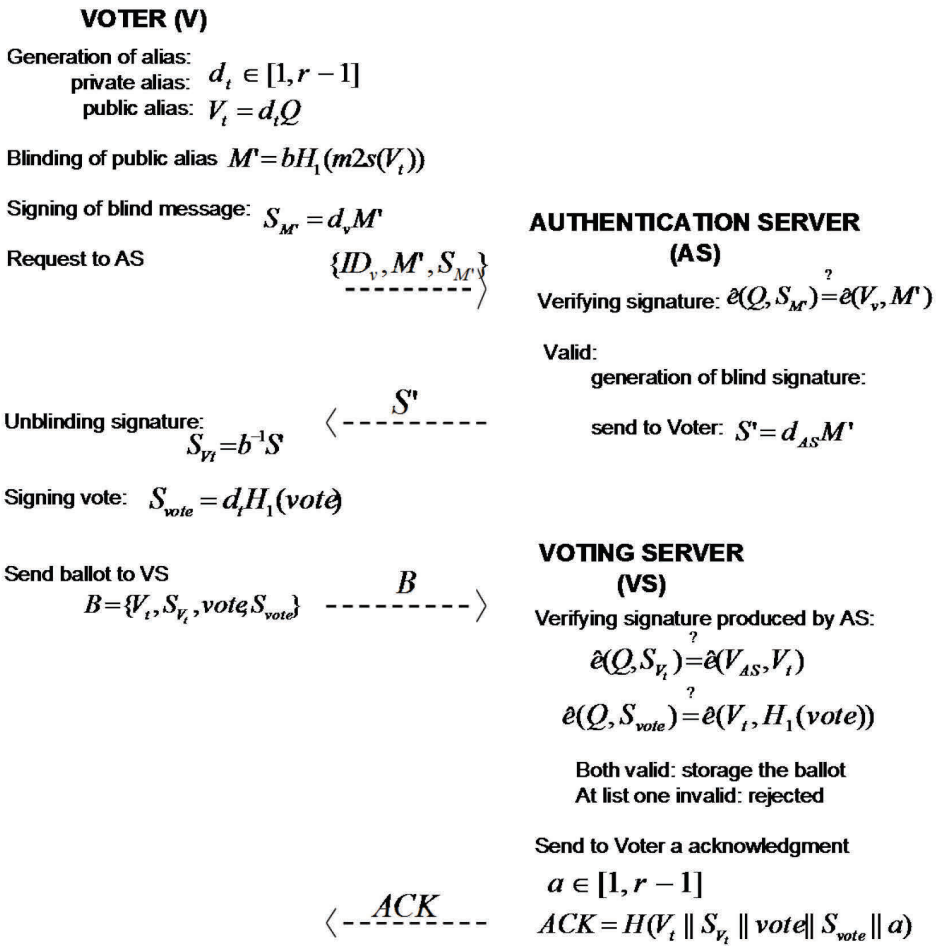


Figure 3 Structure and functionality of the e-voting protocol

Library implementation

Two cryptographic libraries were developed in the Java language, the libraries run on the Servers (AS and VS) and on each Android Smartphone (V). For the reported results, the tests were executed on the following hardware. Each Server is a computer Intel Core i5 @ 2.53 GHz processor, 4 GB RAM, Linux Ubuntu 11.04 O.S. The Smartphone is a Sony Xperia S LT26i, ARMv7 Processor rev 4 @ 1.512 MHz, the O.S. is Android 2.3.3.

Table 1 presents the cryptographic operations that are used by the e-voting protocol with the following parametrization: $G_1=\langle P\rangle$ is an additive group generated by the base point $P\in E(F_p)$; $G_2=\langle Q\rangle$ be a group generated by the base point $Q\in E(F_{p^2})$ and $G_T=\langle\gamma\rangle F^*_{p12}$ be a multiplicative group; all of them of order $|r|$=256 bits.

These results correspond to the average of 100 runs of each operation on the Servers and the Smartphone.

The scalar multiplication was implemented into the groups G_1 and G_2 using the *w-NAF* algorithm [13], obtaining 5.66 *ms* and 621.04 *ms* respectively. The big difference is caused by the length of the elements in G_2 which has points with corresponding polynomial coordinates $x=(a_1\beta+,a_{0|})$ and $y=(b_1\beta+,b_{0|})$ where $|a_1|\approx|a_0|\approx|b_1|\approx|b_0|\approx$ 256 bits.

The bilinear pairing function is the high-speed implementation of *ate pairing* for desktop computers and it was proposed in [14]. It is public available on the Internet.

It is important to notice that the bilinear pairing function is not available in the library that runs on the Android Smartphone, due to this operation is only required on Servers-side, according to the e-voting protocol implemented.

Table 1 CPU Times of cryptographic operations

Operation	*Android Smartphone (voter) (ms)*	*Servers (ms)*
Scalar multiplication on G_1	5.660	0.011
Scalar multiplication on G_2	621.407	0.017
Bilinear pairing function	3502.410	0.031

The developed e-voting system uses the tools described above. As shown in figure 3, the e-voting protocol begins with the ballot generated by the voter V (Android Smartphone) and the AS, followed by the transmission of the ballot to the VS.

The operations performed on the voter side are four scalar multiplications in G_1 and one more in G_2. On Serves-side, two scalar multiplications in G_1, and six bilinear pairing functions are executed, during the authentication and voting phases.

It can be seen that the most expensive operations are not computed by the Android Smartphone, this makes efficient the developed e-voting system.

Table 2 presents the number of cryptographic operations that are required by each entity (Smartphone and Servers) according to the phase of the e-voting protocol.

Table 2 Number of cryptographic operations required for the e-voting system

Phase	*Operation*	*Smartphone (Voter)*	*Servers (AS,VS)*
Authentication	Scalar multiplication on G_1	3	1
	Scalar multiplication on G_2	1	0
	Bilinear pairing function	0	2
Voting	Scalar multiplication on G_1	1	1
	Scalar multiplication on G_2	0	0
	Bilinear pairing function	0	4

Table 3 shows the total CPU time required by the protocol to complete a session between the voter and the Servers.

Table 3 CPU Times of the e-voting in a complete session

Phase	*CPU Time (ms)*
Authentication (Voter-AS)	630.01
Voting (Voter-VS)	6.20
Total:	636.02

It is important to mention that in literature there exist many implementations of the bilinear pairings for desktop computers; however, there are currently a few implementations for mobile devices. A version of bilinear pairings for Smartphone is proposed in [15], where is implemented a cryptographic bilinear pairing library in the Java language for the Windows Mobile O.S. According their results, the CPU times are very competitive for the scalar multiplication on elliptic curves in G_1. However, as can be seen in table 1, the time for this operation in this paper is 5.66 *ms*. Besides, they do not implement a scalar multiplication in G_2, which for the e-voting system implemented, it is quite necessary.

E-voting system in the Android Smartphone

The communication between the voter (Android Smartphone) and the electoral Servers is realized using Jersey framework which is used to provide Web Publication Services and to serialize classes with JSON. These classes are show in figure 4.

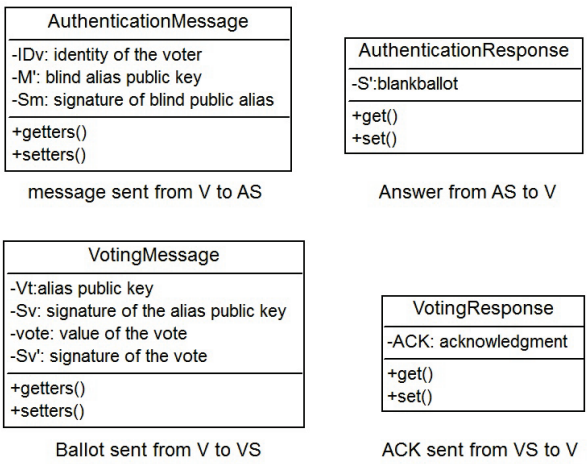


Figure 4 Classes used by the e-voting system

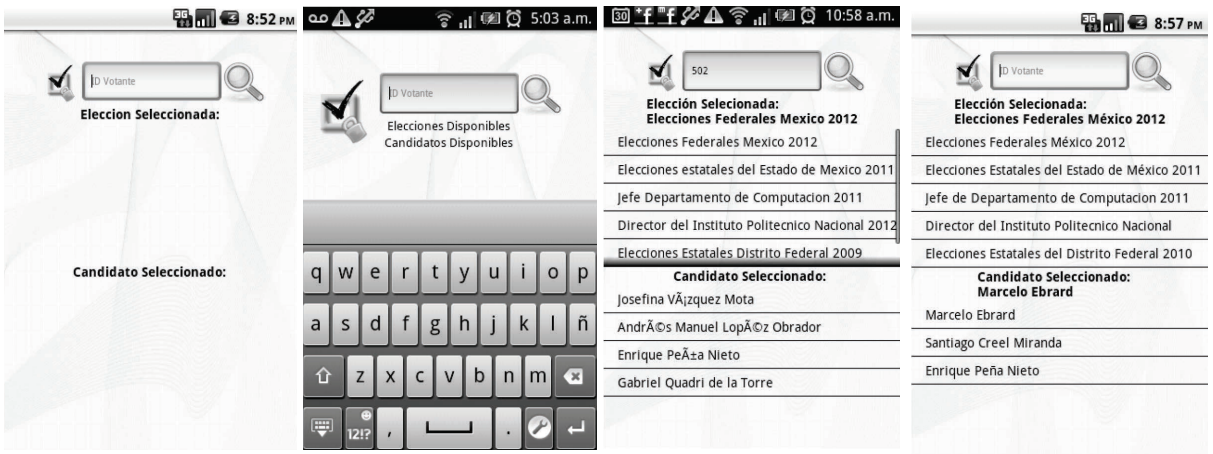
Table 4 shows the transmission times required to develop a complete session among the Servers and the Voter.

F. Smartphone client

The e-voting system starts with the request of the voter to cast the vote. Figure 5 shows the GUI of the system. First, the voter types the ID in the text area control (a). Afterward, the system displays the elections (b) and the candidates (c) using the magnifying glass control, the e-voting protocol starts (d) when the user chooses the candidate.

Table 4 CPU Times of the e-voting in a complete session

Message	*Message length (bits)*	*Transmission time (ms)*
From Smartphone to AS $\{ID_V, M', S_M\}$	{2, (254,1), (254,1)} = 512	5.10
From AS to Smartphone $\{S'\}$	{(254,1)} = 255	4.88
From Smartphone to VS $\{V_t, S_{Vt}, vote, S_{vote}\}$	{(508,1), (254,1), 2, (254,1)} = 1021	20.54
From VS to Smartphone $\{ACK\}$	{128} = 128	2.33



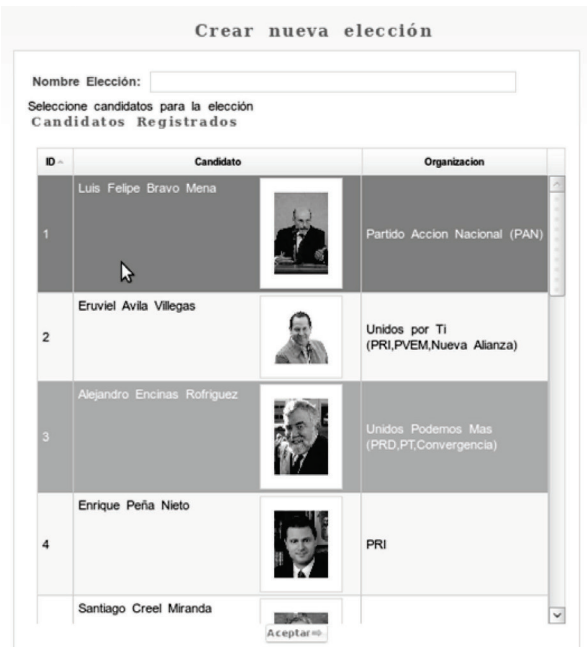
(5.a) (5.b) (5.c) (5.d)

Figure 5 E-voting system graphical user interface in the Smartphone

G. Web application

A Web application is used by the electoral entities to register the election also the candidates. The GUI for the electoral entities is shown in figure 6. It allows the electoral entities to choose the election (a), create a new election (b) registering the name of the election and the number of contenders.

(6.a)



(6.b)

Figure 6 E-voting interfaces in an electoral server

H. Election results

At the end of the election, the results are published in a Web page that can be visited using a Web browser. The results show in figure 7 corresponding to mexican presidential election. The candidates are identified with their pictures and manes. For this example, the winner is the PRI candidate.

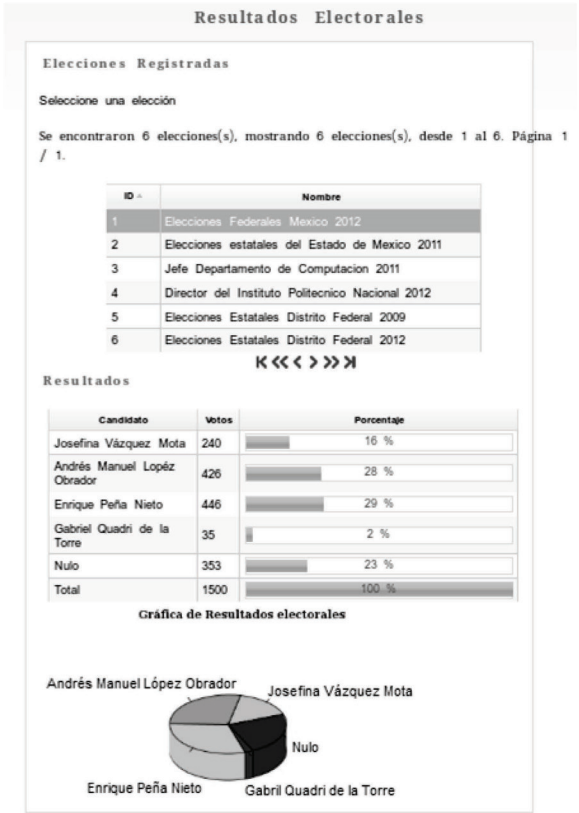


Figure 7 Results of the election

Conclusions

This work presents an e-voting system which uses a secure e-voting protocol based on bilinear pairings. The system allows emitting the vote over the Internet through an Android Smartphone. It was necessary to develop two cryptographic libraries, one for the Servers and other for the Android Smartphone. Both libraries implement the scalar multiplication in the G_1 and G_2 groups, besides of the arithmetic operations of elliptic curves.

The implementation of the library in both entities (Android Smartphone and Servers) guarantee the integrity of the vote and the anonymity of the voter using digital signatures and blind signatures based on bilinear pairings. This is very useful in applications that require to guarantee more security without utilize 1024-bit length numbers at least.

The e-voting system requires only for messages to cast an anonymous vote in an election. In the experiments, the measured transmission times corresponding to these messages, in the order of milliseconds units. The time consumed by cryptographic operations on the Android Smartphone is also very short, compared with other implementations. A plus in efficiency is that the most expensive operation –bilinear pairing- is only executed on Servers-side. Then, the developed e-voting system introduced in this work is reliable and efficient.

References

1. O. Cetinkaya. *Analysis of Security Requirements for Cryptographi.* International Conference on Availability, Reliability and Security. Barcelona, Spain. 2008. pp. 1451-1456.

2. L. Wang, J. Guo, M. Luo. *A More Effective Voting Scheme Based on Blind Signature*. International Conference on Computational Intelligence and Security. Guangzhou, China. 2006. pp. 1507-1510.

3. B. Campbell, C. Tossel, M. Byrne, P. Kortum. *Voting on a Smartphone: Evaluating the Usability of an Optimized Voting System for Handled Mobile Devices*. Human Factors and Ergonomics Society Annual Meeting. Las Vegas, USA. 2011. pp. 1100-1104.

4. M. Esponda. "Electronic voting on-the-fly with mobile devices". *Journal SIGCE Bull ACM.* Volume 40, Issue 3. 2008. pp. 93-97.

5. T. Ahmad, J. Hu, H. Song. *An efficient mobile voting system security scheme based on elliptic curve cryptography.* 3rd International Conference on Network and System Security. Surfers Paradise, Australia. 2009. pp. 474-479.

6. Y. Qiu, H. Zhu. *Somewhat Secure Mobile Electronic-Voting Systems Based on the Cut-and-Choose Mechanism*. Computational Intelligence and Security. Beijing, China. 2009. pp. 446-450.

7. L. López, L. Pérez, F. Rodríguez. *"*A pairing based blind signature e-voting scheme". *The Computer Journal, published online.* doi: 10.1093/comjnl/bxt069. 2013. pp. 1-12.

8. P. Barreto, M. Naehrig. *Pairing-friendly elliptic curves of prime order.* Selected Areas in Cryptography, Montreal, Canada. 2006. pp. 319-331.

9. D. Boneh, M. Franklin. "Identity-Based Encryption from the Weil Pairing". *Journal on Computing*. Vol 32. 2003. pp. 213-229.

10. D. Boneh, H. Shacham, B. Lynn, "Short Signature from the Weil Pairing". Journal of Cryptology. Vol. 14. 2004. pp. 297-319.

11. D. Chaum. *Blind Signatures for Untraceable Payments*. Advances in Cryptology. Ed. Springer Verlag, Santa Barbara California, USA. 1983. pp. 199-203.

12. A. Boldyreva. *Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme*. Theory and Practice in Public Key Cryptography. Miami, Florida, USA. 2003. pp. 31-46.

13. D. Hankerson, A. Menezes, S. Vanstone. *Guide to Elliptic Curve Cryptography*. 1st ed. Ed. Springer-Verlag. New York, USA. pp. 96-141.

14. J. Beuchat, J. González, S. Mitsunari, E. Okamoto, F. Rodríguez, T. Teruya. *High-Speed Software Implementation of the Optimal Ate Pairing over Barreto-Naehrig Curves*. Pairing-based cryptography, Yamanaka Hot Spring, Japan. 2010. pp. 21-39.

15. C. Okida, D. Goya, R. Terada. "Java Cryptographic Library for Smartphone". *Journal Latin America Transactions IEEE*. Vol. 10. 2012. pp.1377-1384.