Implementation of Confidentiality and Anonymity as Services in an E-Voting System for Educational Institutions

Felipe Andrés Corredor Chavarro
Professor, Faculty of Basic Sciences and Engineering, University of the Llanos, Villavicencio, Colombia.

Cristian David Carranza Homes
Systems Engineer, GITECX Research Group, University of the Llanos, Villavicencio, Colombia.

Diana Cristina Franco Mora
Professor, Faculty of Basic Sciences and Engineering, University of the Llanos, Villavicencio, Colombia,

Abstract

Electronic voting (E-voting) systems must meet certain security and usability requirements as well as a sufficient confidence level so that its implementation and usage is welcomed by voters. These systems need to have a robust security scheme able to deal with cyber threats in order to guarantee the same services and functionalities of traditional voting systems. One of the main problems with E-voting implementations is ensuring anonymity during and after the electoral process, which although it seems to have an antagonistic state with other security services also required, such as authentication and non-repudiation, its implementation must be achieved in an articulated manner, so that they become complementary. This paper presents the architecture and development of an alternative E-voting solution for educational institutions (called SIVIT) that complies with the Colombian legal framework; analyzing the effectiveness of its modules of anonymity, confidentiality and auditability of the system, based on the cryptographic and technological mechanisms used.

Keywords: Anonymity, confidentiality, educational institutions, electronic ballot box, telematic voting.

INTRODUCTION

In the information society there has been a rapid and growing tendency for daily activities – carried out by traditional means – to converge towards the use of computational capabilities and telematic networks for a better execution; in this sense, electronic democracy (E-democracy) has been consolidated, as the use of ICT and telematic networks for the improvement of politics and the promotion of citizen participation in democratic processes [1].

E-voting systems, as the greatest expression of E-democracy, are not exempt of the implementation risks of similar systems such as the ones related with E-commerce, E-government and E-learning; this kind of systems must have a robust security scheme able to resist attacks and cyber threats, in order to provide the same functionalities and services of conventional systems, with the same reliability level but with the added value of greater integrity.

According to the latest digital security policy in Colombia, defined in CONPES 3854, more than 65% of security incidents affect the citizens and government sectors (42,4% and 23,9% respectively), which are critical for the implementation of digital democracy. At the same time, there is a legal framework that promotes and supports the implementation of E-voting systems in the country, some examples of this are: Law 1475 of 2011 (Automation of the Colombian electoral process and biometric authentication), the Electronic Commerce Law (which regulates digital signatures and certificates as well as certification authorities), Law 1581 of 2012 (Personal data protection, complemented by decree 1377 of 2013) and Law 1273 of 2009 (Protection of information and data).

Although there are countries where E-voting is not accepted (such as Germany, Finland, Holland, Ireland and the United Kingdom), Colombia has decided to implement it gradually; but there are mechanisms that must be appropriated, that go beyond the legal framework and are more related to technological and security schemes. These mechanisms must be implemented not only for the country's big electoral events, but also from the root of the process that lies in the education and participation of children and young people at the level of educational institutions of middle and high school, where they are allowed to exercise this right, electing their representatives and statutory representations in collegiate bodies.

This kind of systems focus on providing security services such as authentication, confidentiality, information integrity, access control, availability and non-repudiation. However, given the need for keeping secret the voter identity in traditional electoral systems, security schemes for E-voting systems designed to support all the stages of an electoral process must include anonymity as a service as well. Although this last one may appear to have an antagonistic state with other services also required such as authentication (verifying identity) and non-repudiation (proving who made a certain action) it is also an essential security service for these systems.

For the implementation of the security services, different cryptographic schemes are adopted, such as Blind signature, Mixing Networks and ElGamal for anonymity and public-key schemes such as digital signatures and certificates for confidentiality and non-repudiation. In such a way that when these services are implemented in an articulated manner, they stop being antagonistic and become complementary, giving the electoral process the same (or an even higher) degree of transparency and reliability than a traditional system.

As the state of the art suggests, research on electronic voting is still active trough the work of people like Dr. David Chaum who writes on topics such as the protocols being used [2] [13] and different voting schemes based on digital signature mechanisms, Mix networks [34] and cryptosystems with computational complexity of discrete logarithms, through asymmetric encryption algorithms such as RSA, ElGamal, Paillier among others [11], [14], [17], aiming to provide authentication, anonymity, non-repudiation, confidentiality and vote-integrity services; taking advantage of its homomorphic feature to carry out the counting and totalization of results; to such an extent that auditable End-to-End E-voting systems have been created [5], [7], [9], [18], [19], that allow the voter to audit (verify) the validity of the ballot before issuing its voting intention and verify that his vote has been taken into account without revealing the identity of the candidate he voted for, either from home, making an internet query or right in the voting place.

GITECX, a research group on open technologies, developed an E-voting solution alternative (called SIVIT) that complies with the Colombian legal framework and is geared towards educational institutions and in which security mechanisms were implemented in an efficient manner to support electoral processes in these ‘smaller’ scenarios where traditional voting systems, however, would be more expensive.

This paper presents a review of some E-voting related tools and its protection mechanisms, followed by the proposed architecture, where the different logical and physical components of the solution are explained. Finally, test-case results, security-components integration and conclusions are discussed.

SECURITY MECHANISMS IN E-VOTING SYSTEMS

In the global context several electronic voting systems have been implemented to automate all the stages of an electoral process, with Brazil and Venezuela being two of the seven countries in the world that have opted for a full implementation for presidential elections. Smartmatic© solutions were used in these two cases; these solutions have biometric authentication mechanisms for the voter authentication stage and satellite data transmission for the result totalization stage [20]–[24]. It should be mentioned that technical information details are hard to access as these are proprietary technologies.

Telematic voting tools in context

Telematic voting systems share several aspects but should be adapted to different cultures and contexts, which results in different alternatives in logical design and component structure, as proposed in [25]. In that order of ideas, there are different kinds of tools for the purpose: electronic voting systems, telematic voting systems and Internet voting systems.

While each one of them can be seen as technological advancement, they do not necessarily mean social progress, as systems designed to technologically impact the conventional electoral method must surpass a series of security and usability requirements as well as achieve a sufficient confidence degree so that its implementation and usage is welcomed by voters. In **Table 1** some systems that fall within the aforementioned classification are described.

Table 1. Other tools in the context

System	Components	Type	Technology	Scheme/Algorithm	Ref.
Helios voting	SSL connection, authentication, verification, administration, result publication	Telematic (Internet voting)	Software	Sako-Kilian/Benaloh Mixnet, ElGamal	[26],[27], [28]
Wombat Voting System	SSL connection, verification, administration, result publication	Telematic	Software, Hardware	Mixnet y Zero Knowledge, ElGamal	[29],[30].
Scantegrity	Auditing, verification, result publication.	Electronic	Software, Hardware	Zero Knowledge, Mixnet RPC, RSA	[9], [19], [31], [32]

METHODOLOGY

Architecture design

This project proposed the design and construction of a telematic voting tool called SIVIT for which a series of software and hardware based modules were defined and developed, as these were necessary to assume automatically the fundamental features (in functional, legal and information security terms) of a complete electoral process for any educational institution in Colombia.

As shown in **Figure 1**, SIVIT is supported by a client-server architecture running on a server-oriented Unix operating system that guarantees the stability and deployment of telematic services (such as SSH and Postgres), in this case Debian stable 8.3 – Jessie – was selected as the operating system for the server layer, while for the client layer there is the possibility of using other stable Linux distributions or Windows® that support the Java Virtual Machine. The proposed architecture has six modules for the server layer and three for the client one, which are presented as hexagons in the same figure. **Table 2** presents the selected technologies for the implementation of each module.

The system was designed to work in a local network with external security conditions (firewall, intrusion detection, monitoring system, encrypted backups, etc.) that work complementarily in an isolated scenario. The suggested network is wired; however, the voting booths were equipped with a redundant wireless channel for communication with the server.

Table 2. SIVIT Modules

Mod	Cab	Serv.	Tec	Stage
Admin.		X	PHP, postgreSQL	Pre-election day
Result publication		X	PHP, PostgreSQL	Scrutiny
Authentication	X	X	JDK1.8, Cross match SDK-java	Verification of voters identity
Encrypted communication	X	X	JDK 1.8.0 OpenSSL, Apache HTTPClient	Vote registration
Anonymity	X	X	JDK 1.8.0, JCA-API	Voting
Auditing		X	PHP, PostgresQL	Auditing and verification

Physical structure

In SIVIT's architecture design process, three voting booths were designed, with portability, device integration, physical access control security for device operation, and comfort and voter privacy as functionality criteria. In **Figure 2**, the finished voting booth prototype is presented.

Each voting booth was equipped with a PC-type Computer (Intel® Core™ i5-4460S processor, up to 3.40 GHz and 8 GB DDR3-160 RAM memory) and other devices presented in **Figure 2** and described in **Table 3**. Each device intervenes in the functionality of the modules proposed in **Figure 1**; as an example, devices number 2 and 5 in **Table 3**, make the multifactor (fingerprint, student card and User/Password) authentication service possible.

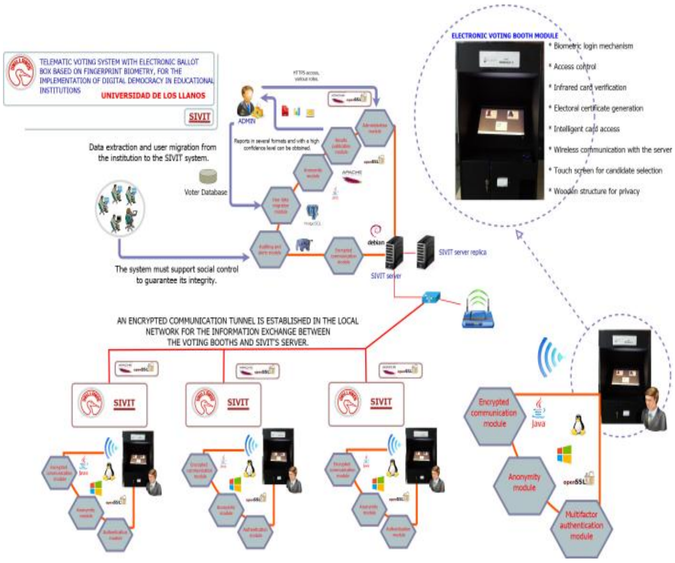

TELEMATIC VOTING SYSTEM WITH ELECTRONIC BALLOT BOX BASED ON FINGERPRINT BIOMETRY, FOR THE IMPLEMENTATION OF DIGITAL DEMOCRACY IN EDUCATIONAL INSTITUTIONS
UNIVERSIDAD DE LOS LLANOS
SIVIT
ADMIN
HTTPS access, various roles.
Reports in several formats and with a high confidence level can be obtained.
Data extraction and user migration from the institution to the SIVIT system.
Voter Database
The system must support social control to guarantee its integrity.
debian
SIVIT server
SIVIT server replica
ELECTRONIC VOTING BOOTH MODULE
* Biometric login mechanism
* Access control
* Infrared card verification
* Electoral certificate generation
* Intelligent card access
* Wireless communication with the server
* Touch screen for candidate selection
* Wooden structure for privacy
AN ENCRYPTED COMMUNICATION TUNNEL IS ESTABLISHED IN THE LOCAL NETWORK FOR THE INFORMATION EXCHANGE BETWEEN THE VOTING BOOTHS AND SIVIT'S SERVER.
SIVIT
SIVIT
SIVIT
Encrypted communication module
Java
openSSL
Anonymity module
Multifactor authentication module

Figure 1. SIVIT's architecture.

Table 2. Devices in voting booths and SIVIT server

No	Device/Component
1	AOC E2060VWT 19,5" touch screen
2	Digital Persona UareU4500 fingerprint reader
3	Electoral certificate dispensing slot (thermal printing)
4	Front door with lock, for maintenance access
5	Card reading / barcode compartment with USB plug & play scanner
6	TP-Link (TL-WN722N) 150 Mbps wireless card
7	Dell Precision T1700, Intel Xeon E3-1200 Server (Application container)/Electronic ballot box
8	Dell UltraSharp U2413 as the Server touch screen
9	Ventilation ducts.

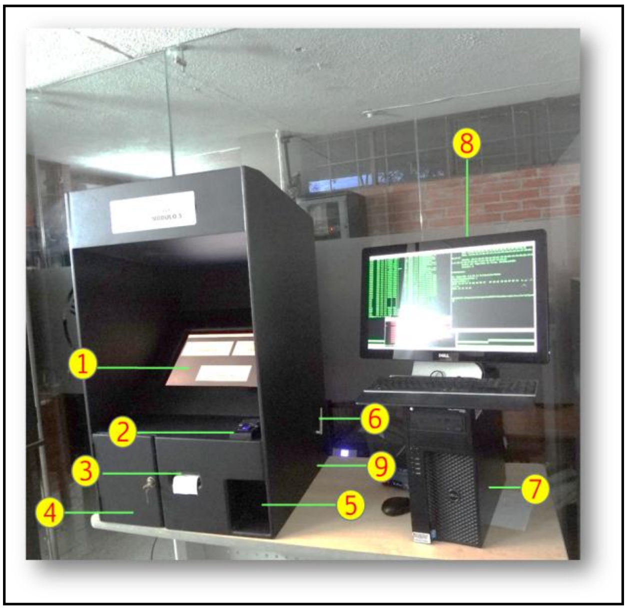


Figure 1. Voting booth and server.

IMPLEMENTATION

Following, procedures and theoretical aspects on which SIVIT's implementation was based are described. An original public-key infrastructure (PKI) was deployed, including all its components (certification authority, registration authority, key deposit, digital certificates and signatures) and ElGamal cryptographic scheme.

Secure communication

Each module's security is based on SIVIT's PKI, which allowed the deployment of the confidentiality service and operates as follows:

Key deposits generation

A C.A. (Certificate Authority) was created for SIVIT general administrator, based on a pair of keys $K_{P_{AC}}$ and $K_{S_{AC}}$ generated with KeyTool; with this same tool, the key deposits and certificates for the tree modules (Cab_i) and the server ($Serv$) were generated:

$$Client \cdot jks = keystore_{Cab_i}(Ks\, cab_i, Kp\, cab_i) \quad (1)$$

$$Server \cdot jks = keystore_{Serv}\left(Ks_{serv}, Kp_{Serv}\right) \quad (2)$$

Where,

$$Cab_i = cabina_i, i \in \mathrm{N} \quad (3)$$

Export of the C.A. certificate

Then, a C.A. digital certificate was exported and used for digitally signing the certificates of the server and each one of the modules:

$$CerTTP.cer = CD_{CA}\left(E_k\left(Ks_{CA}\right), Kp_{CA}\right) \quad (4)$$

Signature request generation

Subsequently, a signature request CSR (Certificate Signing Request) was made by the server and clients to the certification authority

$$Client \cdot csr_i = CSR_{cab_i}\left(E_k\left(Ks_{cab_i}\right), Kp_{cab_i}\right) \quad (5)$$

$$Server \cdot csr = CSR_{cab_i}\left(E_k\left(Ks_{serv}\right), Kp_{serv}\right) \quad (6)$$

C.A. signing process

Corresponding certificates are digitally signed by the C.A., (5) and (6).

$$Client \cdot cer_i = S_{CA}\left(Client \cdot csr_i\right) \quad (7)$$

$$Server \cdot cer = S_{CA}\left(Server \cdot csr\right) \quad (8)$$

Import of certificates signed by C.A.

The C.A. certificate and each one of the client and server certificates signed by the C.A. are imported in the client and server modules, verifying that the process is done cross-wise, so that the C.A. certificate is available both in the client modules (voting booths) and in the server, and at the same time, the server's signed certificate is on each one of the clients, and each one of the clients signed certificates are on the server.

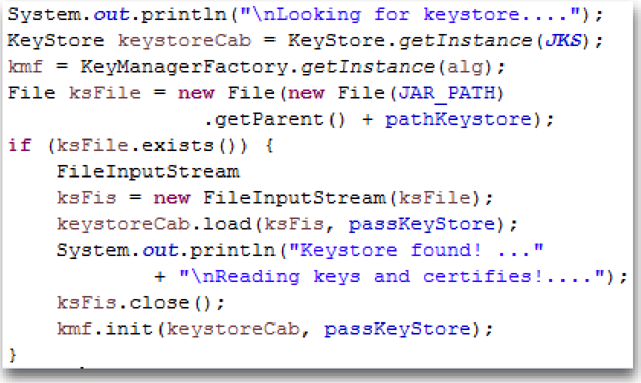
$$Cab_i \leftarrow \left(CerTTP \cdot cer, Server \cdot cer\right) \quad (9)$$

$$Serv \leftarrow \left(CerTTP \cdot cer, Client \cdot cer_i\right) \quad (10)$$

From that moment, SSL socket implementation is made possible, based on the generated digital certificates, which allow for private and binding end-to-end communication (confidentiality and non-repudiation services); in which both voting booths and server are able to request the authentication of the counterparty; shielding the communication channel from interception or modification attacks, as well as from system communication module hijacking of traffic/sessions.

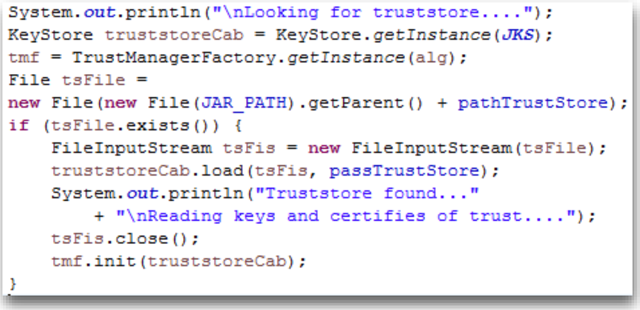
The following Java code excerpts show the implementation of communication security based on digital certificates. Figures 3 and 4 show the loading of the key and certificate files needed for the construction and configuration of the SSL sockets, as shown in Figure 5.

As socket configuration has been completed on both ends for the establishment of the secure channel between clients and server, the first application to run should be the server one, which waits for connection requests from the clients.



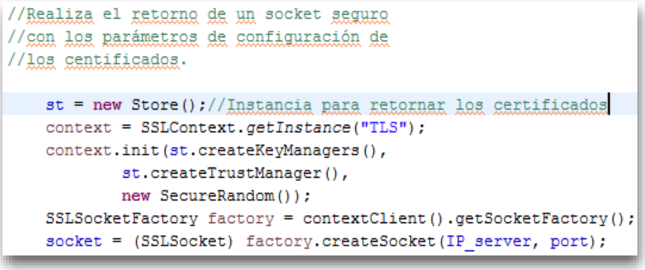
```
System.out.println("\nLooking for keystore....");
KeyStore keystoreCab = KeyStore.getInstance(JKS);
kmf = KeyManagerFactory.getInstance(alg);
File ksFile = new File(new File(JAR_PATH)
                .getParent() + pathKeystore);
if (ksFile.exists()) {
    FileInputStream
    ksFis = new FileInputStream(ksFile);
    keystoreCab.load(ksFis, passKeyStore);
    System.out.println("Keystore found! ..."
            + "\nReading keys and certifies!....");
    ksFis.close();
    kmf.init(keystoreCab, passKeyStore);
}
```

Figure 2. Key deposit reading.



```
System.out.println("\nLooking for truststore....");
KeyStore truststoreCab = KeyStore.getInstance(JKS);
tmf = TrustManagerFactory.getInstance(alg);
File tsFile =
new File(new File(JAR_PATH).getParent() + pathTrustStore);
if (tsFile.exists()) {
    FileInputStream tsFis = new FileInputStream(tsFile);
    truststoreCab.load(tsFis, passTrustStore);
    System.out.println("Truststore found..."
        + "\nReading keys and certifies of trust....");
    tsFis.close();
    tmf.init(truststoreCab);
}
```

Figure 3. Trusted certificates deposit reading.



```
//Realiza el retorno de un socket seguro
//con los parámetros de configuración de
//los centificados.

    st = new Store();//Instancia para retornar los certificados
    context = SSLContext.getInstance("TLS");
    context.init(st.createKeyManagers(),
            st.createTrustManager(),
            new SecureRandom());
    SSLSocketFactory factory = contextClient().getSocketFactory();
    socket = (SSLSocket) factory.createSocket(IP_server, port);
```

Figure 4. SSL socket configuration using digital certificates.

Anonymity

This module is based on ElGamal discrete logarithm encryption scheme, which is used (through an original implementation) to encrypt each vote, store it in the ballot box and generate a code for the electoral certificate, as described below.

Generation of keys and parameters on the server

When the electoral journey starts, the server application generates a random prime number ***p*** with a maximum allowed length of ***n*** with which the finite field Z^+ is established. So that:

$$P \le 2^n, P \in Z^+\left\{ \;\; 3,5,7,11...,2^n \right\} \tag{11}$$

Then, the server generates a pseudo-random number **g** which is used to calculate the key pair (public and private), as shown below.

$$g^{n-1}, g \le p-1 \tag{12}$$

The server randomly chooses its secret key **a** according to those generated values, so that:

$$a \le 2^n \wedge a \le p-1 \tag{13}$$

Then the K parameter is calculated:

$$K = g^a \operatorname{mod}(p) \tag{14}$$

Thus, the $Server_{pub}(K,p,g)$ values will be the server's public key. These values together with n are sent to the voting booths after the authentication process, so that there a new key pair for the votes can be generated before these are sent to the server, which is responsible for storing them in the electronic ballot-box.

Key and parameter generation in the voting booths

When the server application receives the server public key $Server_{pub}(K,p,g)$, it temporarily generates its random private key ***b*** under the condition:

$$b \le 2^n \wedge b \le p-1 \tag{15}$$

Using this value it calculates its public key Y_c, such that:

$$Y_c = g^b \operatorname{mod}(p) \tag{16}$$

Then Y_v is generated, as a result of encrypting the message ***m*** using ***K,*** a parameter defined in the server:

$$Y_v = (k^b * m) \operatorname{mod}(p) \tag{17}$$

Subsequently, the server will get the clear message, with the resulting tuple:

$$m = M_c(Y_c, Y_v) \tag{18}$$

Once ***Mc*** is sent to the server, the private key **b**, which was used to perform the encryption, is destroyed at the voting booth, in order to maintain the anonymity of the sender (voter).

Auditing

SIVIT's auditability is based on the generation of the $Y_C + Y_V$ sum hash using the SHA512 algorithm (at both ends of the communication) for each vote emission. This hash is stored in the database in order to perform a subsequent verification. In this way:

$$Hash(Mc) = Sha512(Y_C + Y_V) \qquad (14)$$

Finally, an electoral certificate is printed. This certificate contains the result of (14), so the voter can verify that its vote has been counted, as this same information is stored in the database and can be consulted.

DISCUSSION

SIVIT has been tested in a controlled scenario which simulates a real university environment (case analysis: University of the Llanos), as the system is designed to operate in a local network under the supervision of trained personnel (complying with each institution's policies) that act as electoral authorities, according to universities autonomy or the regulation of basic and secondary institutions.

Test case

The electoral day of the Faculty of basic sciences and engineering (Facultad de Ciencias Básicas e Ingeniería, FCBI) was simulated; in which the representatives of the different collegiate bodies are elected, and where the possibility of accessing the real data required to feed the system was given.

Table 4. Election identifiers information

Identifier	**Description**	**Database records**	
Profile	Identifies the profile of the candidate in the institution.	1	Student
		2	Graduate
		3	Professor
		4	Administrative
		5	Researcher
Candidate number	Representation of the candidate on the ballot.	1	Ana Bety Vacca
		2	Mónica Silva
		3	Yesica Pérez
		2	Davixon Rojas
		2	Laura Torres
		1	Jackson Rios
Election Id	Election identifier (Primary key)	1	Students' representative to the FCBI board.
		2	FCBI Deanship
		3	Representative of Systems Engineering students to the program committee.
		4	Representative of Electronic Engineering students to the program committee.
		5	Representative of Biology students to the program committee.
		6	Professors' representative to the FCBI board.
Election type	Identifies the position or representation to which the candidate aspires.	1	Spokesperson
		2	Students' representative to the program committee.
		3	Students' representative to the faculty board.
		4	Students' representative to the academic council.
		5	Students' representative to the upper council.
		6	Professors' representative to the faculty board.
		7	Deanship
		8	Professors' representative to the upper council.
		9	Graduates' representative to the upper council.
Day Id	Indicates the election day.	1	Election day for all positions for FCBI students.
		2	Election day for all positions for FCBI professors.

As a sample, the students of the undergraduate programs assigned to the FCBI are assumed. These programs are: Systems engineering, Electronic engineering and Biology (for the latter it is assumed that there are no graduates). In Table 4, the identifiers used to characterize each election stored in the database, as well as its parameterization, are explained (see **Table 5**).

Table 5. Elections defined in the test case

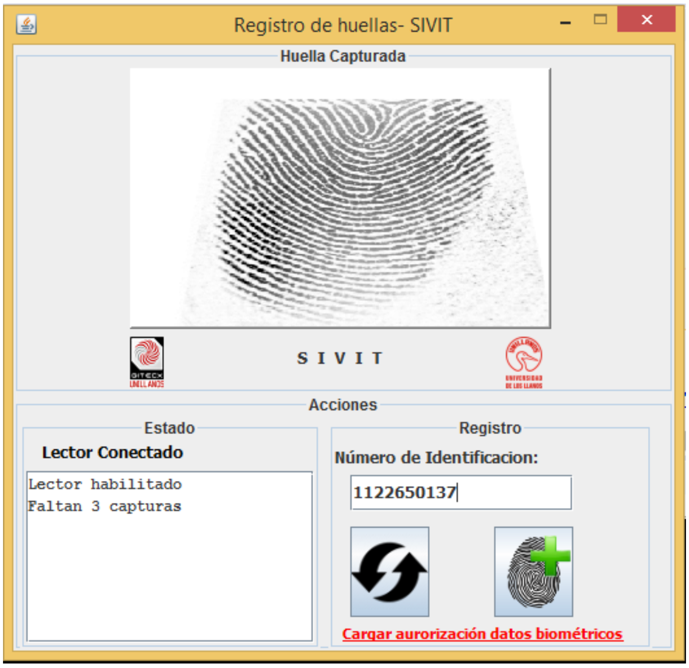
Candidate name	Profile	Candidate number	Election Id	Election type	Day Id
FCBI deanship election					
Candidate 1 – María	3	1	2	7	2
Candidate 2 - Mónica	3	2	2	7	2
Election of the students' representative to the FCBI board					
Jackson Ríos	1	1	1	3	1
Laura Torres	1	2	1	3	1
Yesica Pérez	1	3	1	3	1
Election of the Systems Engineering students' representative to the program committee					
Maicol Parrado	1	1	3	2	1
Davixon Rojas	1	2	3	2	1
Election of the Electronic Engineering students' representative to the program committee					
Dilson Duarte	1	1	4	2	1
Ángela Ramírez	1	2	4	2	1
Election of the Biology students' representative to the program committee					
Maira Pérez	1	1	5	2	1
Melisa Celis	1	2	5	2	1

Electoral process

After the election creation in the system by SIVIT's administrator, once the candidates and the electoral census are uploaded, the system proceeds to load the biometric data authorized (in accordance with the provisions of Law 1581 of 2012 and Decree 1377 of 2013) using a software module developed in Java, which interacts with the biometric reader and provides a user interface that captures the biometric authentication credentials (fingerprint) and password (see **Figure 6**) for its posterior multifactor authentication. Each user enters an identity document and three screenshots of the fingerprint of the right index finger.

In the suffrage stage, the server and voting booths software modules are started. Then the server reads the imported digital certificates, as shown in (9) and (10), which are stored in the key deposits defined in (1) and (2). At the same time, all the parameters needed for the ElGamal based anonymity scheme have been generated (see **Figure 8**), this scheme will be used for all the connections with the voting booths.

When the server receives a connection request, the authenticity of its author is validated using the digital certificates and the secure communication channel is stablished, which ensures that none of the modules from where the votes are sent to the server can be impersonated, nor the data can be discovered, even if the channel is intercepted (as the data is encrypted).


Registro de huellas- SIVIT
Huella Capturada
SIVIT
Acciones
Estado
Registro
Cargar aurorización datos biométricos

Lector Conectado

Lector habilitado
Faltan 3 capturas

Número de Identificacion: 1122650137

Figure 6. Application for capturing the biometric data.

In this way, the voter is authenticated in its corresponding voting booth, entering the access credentials through the user interface shown in **Figure 7** (this can vary according to the type of election**)**. The multifactor authentication process for voters is described in more detail in [33]. Once the authentication is done, the voting booth client application carries out the corresponding visualization controls, in order to show to the voter the dynamic electoral card, which is organized in two ways: the first one with the elections in which the voter can suffrage (see **Figure 9)**, and the second with the candidate list (see **Figure 10)**.

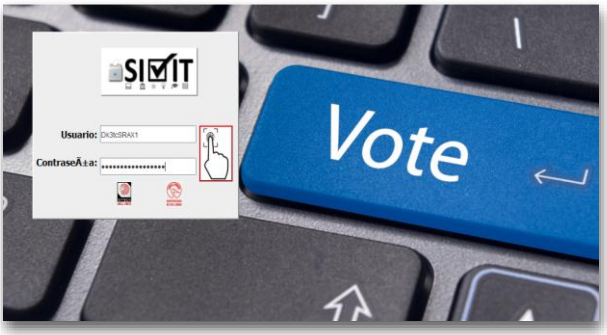
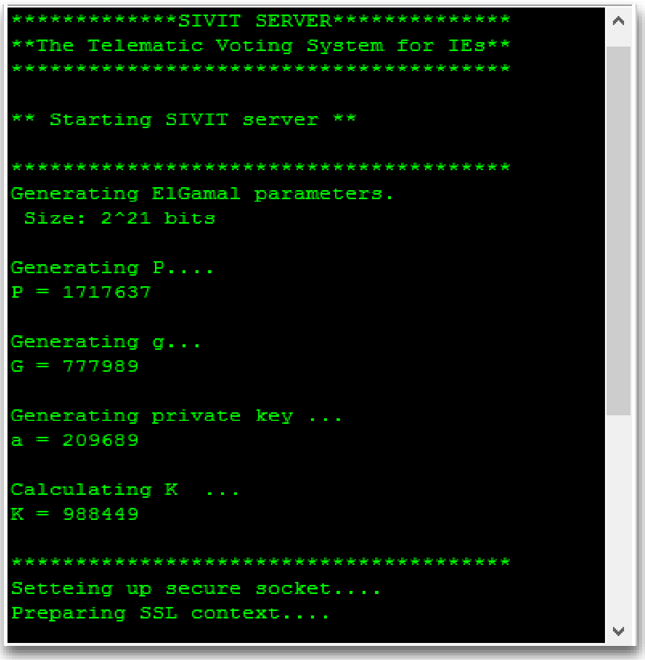


Figure 5. Authentication interface.



```
*************SIVIT SERVER*************
**The Telematic Voting System for IEs**
**************************************

** Starting SIVIT server **

**************************************
Generating ElGamal parameters.
 Size: 2^21 bits

Generating P....
P = 1717637

Generating g...
G = 777989

Generating private key ...
a = 209689

Calculating K  ...
K = 988449

**************************************
Setteing up secure socket....
Preparing SSL context....
```

Figura 6. SIVIT server running.

In this aspect, we worked with the JSON format based on objects from the java-json library and the HashMap class of the "util" java package, in order to take advantage of its << key, value >> property to identify the type of information that arrives at the ends of the communication between each booth and the server.

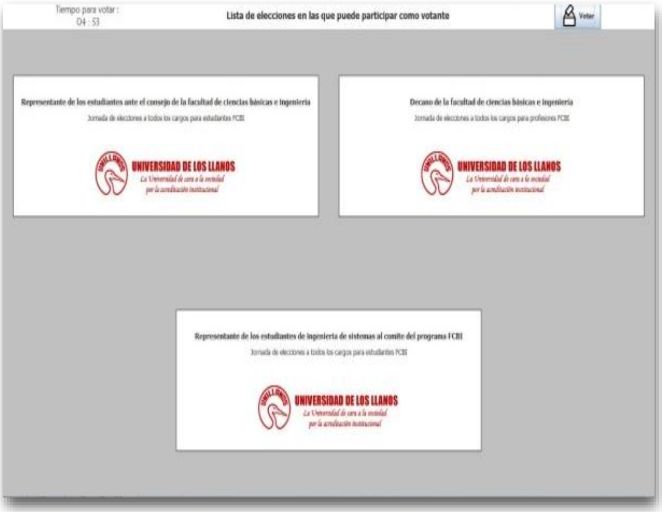

Tiempo para votar : 04 : 53
Lista de elecciones en las que puede participar como votante
Votar
Representante de los estudiantes ante el consejo de la facultad de ciencias básicas e ingeniería
Jornada de elecciones a todos los cargos para estudiantes FCBI
Decano de la facultad de ciencias básicas e ingeniería
Jornada de elecciones a todos los cargos para profesores FCBI
Representante de los estudiantes de ingeniería de sistemas al comite del programa FCBI
Jornada de elecciones a todos los cargos para estudiantes FCBI
UNIVERSIDAD DE LOS LLANOS

Figure 7. Electoral ballot showing the elections in which the voter can participate.

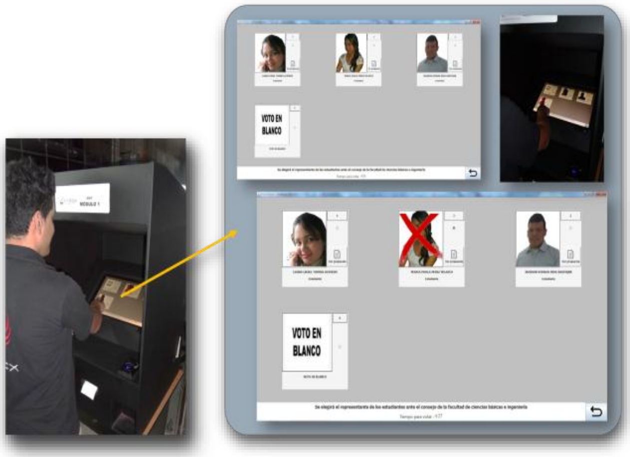


Figure 8. Marking of vote on the ballot.

Once the voter confirms all the votes marked on the corresponding elections, then the anonymity protocol is executed which encrypts the votes and stores them in a database (electronic ballot box). In **Table 6** the anonymity protocol is explained in a more detailed way (based on the theoretical aspects presented in section 4.2 of this article), specifically for the 'Students' representative to the FCBI board.' election, analyzing the process for when the voter confirms a vote for candidate number 3, Yésica Pérez, according to **Table 5**. For test case effects, these were the server parameters generated:

- p = 1624201
- g = 652243
- k = 872446

Table 3. Anonymity protocol in action.

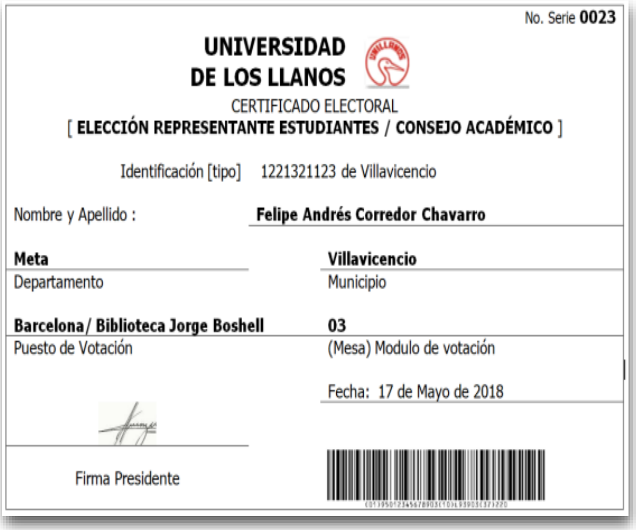
#	Mod.	Process
1	Voting booth	Generates a temporary private key b=1544051 and calculates $\boldsymbol{Y_c} = 113531$ according to (15) and (16) respectively.
2	Voting booth	Computes $\boldsymbol{Y_v}$ = 1452305 according to (17) which is the result of encrypting the clear message **m** = "13" (a codification obtained by concatenating the **Candidate number** and **Election Id** identifiers), in order to minimize the possibility of the same voter generating equal $\boldsymbol{Y_v}$ in different elections; which would happen if **Candidate number** was the only parameter needed to encrypt the message.

3	Voting booth	Calculates **Hash(m)** according to equation (19). Generates a JSON object that contains $\boldsymbol{M_c}$ (see equation (18)) and sends it to the server. Subsequently, **b** is destroyed.
		Example:
		{
		"SaveVotes":
		[
		{ "Yc" : 113531 , "Yv": 1452305 , "ElectionId" : 1},
		{ "Yc" : 113531 , "Yv": ???????? , "ElectionId" : 2},
		{ "Yc" : 113531 , "Yv": ???????? , "ElectionId" : 3}
		]
		}

4	Server	Reads the JSON object which contains $\boldsymbol{M_c}$, performs the inverse process to $\boldsymbol{Y_v}$, using **Yc** and its own private key in order to get **m** = "13" according to equation (18).
5	Server	Computes **Hash(m)** according to equation (19) and stores all the parameters (**Yc,Yv** y **Hash(m)**) in the database electronic ballot box.
6	Voting booth	Reads the JSON object and generates a barcode with **Hash(m)**=1728C358D6A70C372E3A739D61B8342788E7C4FEE02DF8DCE1501D1CE315BCB31C7187 CA3B2336A3E51EF6BFA74747AA781E6D9B3489BF85BDA778DD1407561B
7	Voting booth	Prints an electoral certificate with the barcode generated in the previous step and other information shown in **Figure 10**.
8	Voting booth	Destroys all the instances of the application created to show the ballots and returns to the authentication interface. See **Figure 8**.

All information exchange between the voting booths and the server works similar to a web service, as the protocol is based on actions (represented by each JSON object's primary key) requested at the moment and the information delivered by the server on each action is seen as a resource stored in the database, which is accessed with input data. As an example, the action needed to know the list of candidates among which a voter can choose, travels in the main key of the JSON object as "FindCandidates" and the required input data is the voter's identification number. In this way, the voter's identity is kept secret, even though the authentication process was carried out. Likewise, the voter or a third party (electoral authority) can verify that the vote has been stored in the ballot box, comparing the hash printed on the electoral certificate with the one stored in the server database, without having to reveal for whom the elector voted.

It can be checked in a simple way that with each private key, randomly generated for each session, $\boldsymbol{Y_c}$ and $\boldsymbol{Y_v}$ also change, making it harder to know the identity of the voter. For the purposes of traceability to the protocol operation, 21 bit numbers were used, with the possibility of expanding it in the production environment, in order to increase computational complexity against brute force attacks and expand the finite field to reduce the probability of randomly obtaining the same private keys.



No. Serie **0023**

UNIVERSIDAD DE LOS LLANOS

CERTIFICADO ELECTORAL

[**ELECCIÓN REPRESENTANTE ESTUDIANTES / CONSEJO ACADÉMICO**]

Identificación [tipo] 1221321123 de Villavicencio

Nombre y Apellido : **Felipe Andrés Corredor Chavarro**

Meta Departamento	**Villavicencio** Municipio
Barcelona/ Biblioteca Jorge Boshell Puesto de Votación	**03** (Mesa) Modulo de votación
	Fecha: 17 de Mayo de 2018
Firma Presidente	

Figure 9. Electoral certificate.

CONCLUSIONS

Although E-voting systems usually work in controlled and isolated environments, it is important to implement backup actions that support the security mechanisms, in order to keep the system's functionality even if one of the latter fails.

The use of crossed-out digital certificates (server and voting booths) within the public key infrastructure of an E-voting system greatly increases the confidentiality of communication, decreasing the chances of attack to the encrypted channel and the digital signatures from any end, providing added value to security.

The anonymity and non-repudiation services can be interpreted as antagonistic; however, it is possible to implement them both in an E-voting system, thanks to asymmetric cryptography mechanisms, in such a way that they work complementarily and provide guarantees of confidence and transparency equal or higher to those of traditional voting systems.

Using hash functions with a large bit length and the ElGamal cryptographic scheme to carry out the vote encryption process, allowed the proposal of a reliable protocol that guarantees the anonymity of voters, the credibility and auditability of the system (Verification of votes with anonymity guarantee), using for this last the homomorphic property of the cryptographic scheme.

ACKNOWLEDGEMENTS

The authors express their gratitude to God and their families for always supporting this project. In the same way, it is important to note that this work would not have been possible without the financial support of the University of the Llanos.

REFERENCES

[1] J. C. Gallardo, "TASSI 2014 Posibilidades del voto telemático en la democracia digital," p. 37, 2014.

[2] W.-C. K. Sheng-De, W, "A secure and practical electronic voting scheme," *Comput. {&} Secur.*, vol. 23, no. 23, pp. 330–337, 2004.

[3] D. Chaum, "Secret-ballot receipts: true voter-verifiable elections," *IEEE Secur. Priv. Mag.*, vol. 2, no. 1, pp. 38–47, Jan. 2004.

[4] R. Peter.Y.A., "A Variant of the Chaum Voter-verifiable Scheme," UK, 2004.

[5] R. L. Rivest, "The ThreeBallot Voting System," in *in WPES 2006 — ACM Workshop on Privacy in the Electronic Society*, Cambridge: ACM, 2006, pp. 29–40.

[6] B. Adida and R. L. Rivest, "Scratch &amp; Vote Self-Contained Paper-Based Cryptographic Voting," *Massachusetts Institute of Technology*, Cambridge, p. 11, Oct-2006.

[7] A. Essex, J. Clark, R. T. Carback Iii, and S. Popoveniuc, "Bulletproof Electronic Voting IEEE Spectrum The Punchscan Voting System VOCOMP Competition Submission," Washintong D.C, 2007.

[8] R. L. Rivest and W. D. Smith, "Three Voting Protocols: ThreeBallot, VAV, and Twin," in *Proceedings of USENIX/ACCURATE Electronic Voting Technology (EVT*, Cambridge: press, 2007.

[9] D. Chaum, A. Essex, R. T. Carback III, J. Clark, S. Popoveniuc, A. T. Sherman, and P. Vora, "scantegrity: e nd-to- end voter- verifiable o ptical-scan v oting," *IEEE Comput. Soc.*, 2008.

[10] P. Y. A. Ryan, D. Bismark, J. Heather, S. Schneider, and Zhe Xia, "PrÊt À Voter: a Voter-Verifiable Voting System," *IEEE Trans. Inf. Forensics Secur.*, vol. 4, no. 4, pp. 662–673, Dec. 2009.

[11] F. Song and Z. Cui, "Electronic Voting Scheme About ElGamal Blind-signatures Based on XML," *Procedia Engineering*, vol. 29. Elsevier, pp. 2721–2725, 2012.

[12] H. Pan, E. Hou, and N. Ansari, "E-NOTE: An E-voting system that ensures voter confidentiality and voting accuracy," in *2012 IEEE International Conference on Communications (ICC)*, 2012, pp. 825–829.

[13] M. F. M. Mursi, G. M. R. Assassa, A. A. Abdelhafez, and K. M. Abosamra, "A Secure and Auditable Cryptographic-Based e-Voting Scheme," in *2015 Second International Conference on Mathematics and Computers in Sciences and in Industry (MCSI)*, 2015, pp. 253–262.

[14] V. R. L. Shen, Y. F. Chung, T. S. Chen, and Y. A. Lin, "A BLIND SIGNATURE BASED ON DISCRETE LOGARITHM PROBLEM," *Int. J. Innov. Comput.*, vol. 7, no. 9, pp. 5403–5416, 2011.

[15] T. ElGamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” *IEEE Trans Inf Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[16] C.-L. Chen, Y.-Y. Chen, J.-K. Jan, and C.-C. Chen, “A Secure Anonymous E-Voting System based on Discrete Logarithm Problem,” *Appl. Math. Inf. Sci*, vol. 8, no. 5, pp. 2571–2578, 2014.

[17] E. Mohammed, A. E. Emarah, and K. El-Shennawy, “A blind signature scheme based on ElGamal signature,” in *IEEE/AFCEA EUROCOMM 2000. Information Systems for Enhanced Public Safety and Security (Cat. No.00EX405)*, 2000, pp. 51–53.

[18] P. Y. A. Ryan, D. Bismark, J. Heather, S. Schneider, and Zhe Xia, “Pr{Ê}t {À} Voter: a Voter-Verifiable Voting System,” *IEEE Trans. Inf. Forensics Secur.*, vol. 4, no. 4, pp. 662–673, 2009.

[19] D. Chaum, R. Carback, J. Clark, A. Essex, S. Popoveniuc, R. L. Rivest, P. Y. A. Ryan, E. Shen, and A. T. Sherman, “Scantegrity II: End-to-End Verifiability for Optical Scan Election Systems using Invisible Ink Confirmation Codes,” *USENIX/ACCURATE EVT*, 2008.

[20] Smartmatic, “Testimoniales - Smartmatic,” *Testimoniales*, 2015. [Online]. Available: http://www.smartmatic.com/es/testimoniales/. [Accessed: 11-Jun-2016].

[21] E. J.-G. V. Departamento de seguridad, “Voto electrónico en el mundo. Países con implantación,” 2016. [Online]. Available: http://www.euskadi.eus/botoelek/otros_paises/ve_mundo_impl_c.htm. [Accessed: 10-Aug-2016].

[22] E. J.-G. V. Departamento de seguridad, “Voto electrónico en el mundo. Países en estudio o implantación parcial,” 2016. [Online]. Available: http://www.euskadi.eus/botoelek/otros_paises/ve_mundo_est_c.htm. [Accessed: 16-March-2017].

[23] E. J.-G. V. Departamento de seguridad, “Voto electrónico en el mundo. Legalmente prohibido o paralizado,” 2016. [Online]. Available: http://www.euskadi.eus/botoelek/otros_paises/ve_mundo_paralizado_c.htm. [Accessed: 16-March-2017].

[24] O. Voto electrónico, “Voto electrónico,” 2016. [Online]. Available: http://www.voto-electronico.org/index.php/voto-electronico-en-latinoamerica. [Accessed: 08-Aug-2017].

[25] J. Carracedo Gallardo, *Seguridad en Redes Telemáticas*. Reading, MA: Addison-Wesley, 2014.

[26] A. Ben, de M. Olivier, and P. Olivier, “Helios Voting — Privacy.” [Online]. Available: https://vote.heliosvoting.org/privacy. [Accessed: 21-Sep-2016].

[27] B. Adida, “Helios: Web-based Open-Audit Voting,” in *Proceedings of the 17th Conference on Security Symposium*, Berkeley: USENIX Association, 2008, pp. 335–348.

[28] B. Adida, O. De Marneffe, O. Pereira, and J.-J. Quisquater, “Electing a University President using Open-Audit Voting: Analysis of real-world use of Helios,” 2009.

[29] A. Rosen, A. Ta-shma, B. Riva, and J. Ben-Nun, “Wombat Voting System,” 2011. [Online]. Available: https://wombat.factcenter.org/. [Accessed: 14-Sep-2016].

[30] E. Grundland, “An Analysis of the Wombat Voting System Model,” 2012.

[31] D. Chaum, R. T. Carback, J. Clark, A. Essex, S. Popoveniuc, R. L. Rivest, P. Ryan, E. Shen, A. T. Sherman, and P. L. Vora, “Scantegrity II: End-to-End Verifiability by Voters of Optical Scan Elections Through Confirmation Codes,” *IEEE Trans. Inf. Forensics Secur.*, vol. 4, no. 4, pp. 611–627, Dec. 2009.

[32] A. T. Sherman, R. A. Fink, R. Carback, and D. Chaum, “Scantegrity III: Automatic Trustworthy Receipts, Highlighting Over/Under Votes, and Full Voter Verifiability,” *Proc. 2011 Conf. Electron. Voting Technol. Trust. Elections*, pp. 7–23, 2011.

[33] C. D. Carranza, F. A. Corredor, J. F. Melo, and D. C. Franco, “SISTEMA DE VOTO TELEMÁTICO CON URNA ELECTRÓNICA PARA IMPLEMENTAR DEMOCRACIA DIGITAL EN INSTITUCIONES EDUCATIVAS.” Comité. Congreso Internacional de Ciencias básicas e ingeniería, Villavicencio, p. 7, 2016.

[34] K. M. Abosamra, A. A. Abdelhafez, G. M. R. Assassa, and M. F. M. Mursi, “Journal of Information Security and Applications A practical , secure , and auditable e-voting system,” vol. 36, pp. 69–89, 2017.