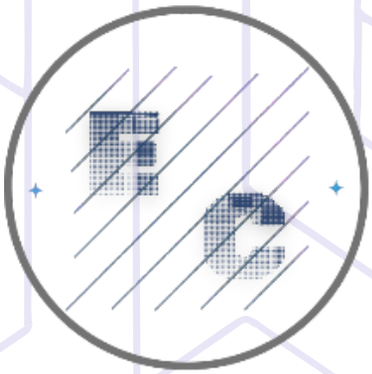
UNIVERSIDAD DE SONSONATE
USO
SCIENTIAE ET BONIS ARTIBUS



MULTIDISCIPLINARY JOURNAL EPISTEMOLOGY OF THE SCIENCES

Universidad de Sonsonate

Volume 3, Issue 3

July–September 2026
Special Edition

CROSSREF PREFIX DOI: 10.71112

ISSN: 3061-7812, www.omniscens.com

Multidisciplinary Journal Epistemology of the Sciences

Volume 3, Issue 3
July–September 2026, Special Edition

Quarterly Publication
Made in Mexico

The Multidisciplinary Journal Epistemology of the Sciences accepts submissions from any field of knowledge, promoting an inclusive platform for the discussion and analysis of epistemological foundations across various disciplines. The journal invites researchers and professionals from fields such as the natural sciences, social sciences, humanities, technology, and health sciences, among others, to contribute with original articles, reviews, case studies, and theoretical essays. With its multidisciplinary approach, it aims to foster dialogue and reflection on the methodologies, theories, and practices that underpin the advancement of scientific knowledge in all areas.

Contact: admin@omniscens.com

9773061781003

Legal Disclaimer

Multidisciplinary Journal Epistemology of the Sciences Vol. 3, Issue 3, July-September 2026, Special Edition, is a quarterly publication edited by Dr. Moises Ake Uc, C. 51 #221 between 16B and 16C, Mérida, Yucatán, Mexico, C.P. 97144, Tel. 9993556027, Web: https://www.omniscens.com, admin@omniscens.com. Responsible Editor: Dr. Moises Ake Uc. Rights Reservation No. 04-2024-121717181700-102, ISSN: 3061-7812, both granted by the Instituto Nacional del Derecho de Autor (INDAUTOR). Responsible for the last update of this issue: Dr. Moises Ake Uc, last modification date: July 1, 2026.

Multidisciplinary Journal Epistemology of the Sciences

Vol. 3, Issue 3, 2026, July–September, Special Edition

DOI: https://doi.org/10.71112/f449y553

ELECTRONIC VOTING APPLICATION USING PAILLIER'S CRYPTOGRAPHIC ALGORITHM

APLICACIÓN DE VOTACIONES ELECTRÓNICAS UTILIZANDO EL ALGORITMO CRIPTOGRÁFICO DE PAILLIER

Álvaro Hernán Zavala Ruballo

El Salvador

DOI: https://doi.org/10.71112/f449y553

Electronic voting application using paillier's cryptographic algorithm

Aplicación de votaciones electrónicas utilizando el algoritmo criptográfico de paillier

Álvaro Hernán Zavala Ruballo[a,*]

alvarohz@usonsonate.edu.sv

https://orcid.org/0000-0003-0715-6596

*Corresponding author: alvarohz@usonsonate.edu.sv, [a]Universidad de Sonsonate, El Salvador.

ABSTRACT

Electoral processes in democratic countries are of vital importance, and despite the benefits offered by technology to make them more efficient and immediate, most people continue to opt for traditional electoral processes using physical ballots and going in person to the polls to vote instead of electronic systems via the Internet. The operationalization is based on the use of secure encryption algorithms and error-free coding implementation, errors that may lead to vulnerabilities. In this article, an electronic election scheme based on homomorphic encryption for vote encryption, vote counting and publication of election results is discussed, given the features of this type of encryption. A web application is developed as an example applying the Paillier algorithm to simulate an electronic election to validate the calculations, allowing the total of votes through the multiplication of cyphertexts obtained with Paillier, and thus to ensure efficiency in the process, integrity and privacy of the data.

Keywords: Electronic election; Homomorphic encryption; Paillier; eVoting; Privacy

DOI: https://doi.org/10.71112/f449y553

RESUMEN

Los procesos electorales en los países democráticos son de vital importancia y, a pesar de los beneficios que ofrece la tecnología para hacerlos más eficientes e inmediatos, la mayoría de las personas sigue optando por los procesos electorales tradicionales mediante el uso de boletas físicas y asistiendo en persona a las urnas para votar, en lugar de utilizar sistemas electrónicos a través de Internet. La operacionalización se basa en el uso de algoritmos de cifrados seguros y en una implementación de código libre de errores, ya que estos pueden dar lugar a vulnerabilidades. En este artículo se analiza un esquema de elección electrónica basado en el cifrado homomórfico para el cifrado de votos, el conteo de votos y la publicación de los resultados electorales, dadas las características de este tipo de cifrado. Se desarrolla una aplicación web como ejemplo aplicando el algoritmo de Paillier para simular una elección electrónica con el fin de validar los cálculos, lo que permite obtener el total de los votos a través de la multiplicación de los textos cifrados obtenidos con Paillier y, de este modo, garantizar la eficiencia en el proceso, así como la integridad y la privacidad de los datos.

Palabras clave: Elección electrónica; Cifrado homomórfico; Paillier; Voto electrónico; Privacidad.

Recibido: 16 febrero 2026 | Aceptado: 30 junio 2026 | Publicado: 1 julio 2026

INTRODUCTION

Electronic election is the application of electronic technology to cast and count votes in an election (Collins English Dictionary, 2024). Typically, the systems that implement it handle voter registration, vote entry, vote casting, vote encryption, ballot transmission to the server, vote storage, vote counting, and tabulation of the election result.

In general, two main types of electronic voting can be identified: Electronic voting

physically supervised by representatives of governmental or independent electoral authorities (e.g., electronic voting machines located in polling stations, also called e-voting); Remote electronic voting through the Internet (also called i-voting), where the voter sends his vote electronically to the electoral authorities, from any location.

This paper develops an electronic election application based on homomorphic encryption for vote encryption, vote counting and publication of election results by applying the Paillier algorithm, allowing the addition of votes through the multiplication of the ciphertexts obtained with Paillier, and thus ensuring efficiency in the process, integrity and privacy of the data.

The application serves as an example to simulate an electronic election in order to validate the calculations with Paillier homomorphic cryptography.

Literature review

Electronic voting has been the subject of study since around the 2000s, in countries such as Switzerland (Haines et al., 2020), and many other nations have attempted it. In many cases, they proceed without giving much thought to whether their electronic voting system is truly secure or could be compromised to subvert democracy. It is not enough to simply trust the word of the software provider.

One of the main obstacles to its adoption is the challenge of guaranteeing all the requirements that traditional voting is based on: authenticity, privacy, integrity, anonymity, auditability, usability, robustness, scalability, effectiveness, fairness, uniqueness, accessibility, and accuracy (Peng, 2011).

The reality is that the internet remains insecure and hackable, which is why security experts have found vulnerabilities in all online voting attempts, including systems in Australia, Estonia, Switzerland, Russia, and the United States.

In the case of Australia, as mentioned in (Halderman & Teague, 2015), votes are encrypted using a "digital envelope" consisting of a randomly generated symmetric key, encrypted once

each with the ElGamal public keys of the election and verification servers. Additionally, the voting choices are encrypted with AES1 using the symmetric key. This has implications for both privacy and system integrity, since both ElGamal and AES require decrypting the content, exposing both the identity and the vote of each voter.

In Estonia, citizens' IDs are smart cards that allow cryptographic operations such as signing and authentication. In (Springall et al., 2014) it is mentioned that (digital) ballots for each vote are encrypted, separated from the signatures, and copied to an isolated machine before being decrypted and counted. This tallying server has access to the full association between the encrypted ballots and the plaintext votes. An attacker who can smuggle this information out through a covert channel could compromise the secrecy of each voter's ballot and even delete votes.

In Switzerland (Zetter, 2019), an online voting system was implemented which included a Zero-Knowledge Proof (ZKP) to validate the election, this concept is explained further in the source. The mistake made by Switzerland would not only allow an attacker to swap all the ballots but also to do so while the zero-knowledge proof still showed that everything was functioning correctly and valid.

Russia, another important case analyzed in (Gaudry & Golovnev, 2020), did not publish technical specifications, but the source code was made available online one day before the first public test. It turned out that it used ElGamal encryption with keys under 256 bits; the encryption was applied three times with different keys, and the designers did not know that triple encryption does not strengthen ElGamal the way it does with DES 2. The first attack involved a simple key recovery, as the CADO-NFS3 library could perform discrete logarithms on a laptop in ten minutes. Electoral authorities then switched to 1024-bit ElGamal, after which a second attack

[1] Advanced Encryption Standard
[2] Data Encryption Standard
[3] It is a software library for factoring integers and computing discrete logarithms in finite fields.

was discovered: a one-bit leak from a subgroup attack “sufficient to distinguish between the two candidates in the election.” Developers denied the attack was effective, but quietly changed the code anyway.

At the Massachusetts Institute of Technology (MIT) in the United States, a security analysis was conducted on the Voatz app (Abby Abazorius, 2020), a mobile voting app used in West Virginia and other states. The app had fundamental security flaws that could allow someone to view and intercept votes as they are transmitted from mobile phones to the company’s server. Among other things, researchers found that even though the traffic from the voter's phone to the server is encrypted, an attacker who can intercept that traffic can still determine how the user voted and thus block any vote that is not for the attacker’s preferred candidate.

It is worth noting that the issues found in each implementation are numerous, and for more detailed information, it is recommended to consult the referenced bibliography.

Theoretical framework

This section discusses the basis for the development of the proposal.

Preliminaries

Electronic Election: Term that was proposed by David Chaum in 1981 (Chaum, 1981) and means the use of technologies to support the voting process of states or organizations to make them more efficient.

Many electronic voting schemes have been proposed (Cedillo, 2017), among the main ones are:

- Blind signatures
- Homomorphic encryption
- Blockchain e-voting
- Mixnet

The first three, use cryptography to solve voting requirements, while the fourth is based on the combination of cryptography with a network infrastructure profusely configured for it. It is worth mentioning that they can be used in combination with Mixnets such as TOR (The Onion Routing) to ensure the anonymity of vote submission.

Cryptography: Is one of the most important fields of information security. It is a method of transferring private information and data over an insecure communication network. Cryptography provides many services such as: confidentiality, authentication, integrity, non-repudiation and accessibility (Menezes et al., 1997).

The solution of the mathematical problems on which they are built is infeasible as long as an appropriate key size is selected (Drummond, 2024).

Cryptography also provides security through its cryptographic primitives such as asymmetric encryption, symmetric encryption, hash functions, key exchange, digital signatures, etc.

Symmetric cryptography: Algorithms that use the same cryptographic keys for both encryption and decryption (Kartit et al., 2016).

The keys in question, represent a shared secret and their security depends largely on the size chosen for them, as well as the safeguard to keep them private between the parties sharing the secret.

The most widely used algorithm in symmetric cryptography is the AES (Advanced Encryption Standard) created by NIST(NIST, 2023a).

Asymmetric cryptography: Public-key (asymmetric) algorithms use a pair of keys (a public key and a private key) and use a different one for each of the two cryptographic operations (e.g., encryption and decryption, or signature creation and signature verification) (Shirey, 2007).

Asymmetric algorithms have key management advantages over equivalent symmetric ones. First, one key of the pair does not need to be known to anyone other than its owner; so, it can more easily be kept secret. Second, although the other key is shared by all entities using the algorithm, it is not necessary to keep that key secret from other entities that do not use it; therefore, key distribution can be more easily accomplished.

One of the most widely used algorithms in the market is the RSA proposed in (Rivest et al., 1978) and from which other schemes such as digital signature is also derived.

Hash Function: A hash function is a computationally efficient function mapping binary strings of arbitrary length to binary strings of some fixed length, called hash-values (Menezes et al., 1997). Some of the cryptographic uses of hash functions are digital signatures and data integrity.

The Secure Hash Standard (SHS) describes widely used and secure algorithms such as SHA-2 (NIST, 2015).

Digital Signature: The digital signature is a cryptographic transformation that provides security services to verify the authentication of origin, data integrity and non-repudiation of the signer (NIST, 2023b).

The private key is used by the signer and the public key by anyone to verify the signature.

Electronic Voting Scheme

The structure of electronic voting systems consists of three phases (OASIS, 2011), pre-election, election and post-election. The processes in the pre-election phase include candidate registration, candidate list computation, voter registration (validated with the electoral roll), and eligible voter list computation. Eligible voters cast their votes during the election phase and each vote is encrypted for storage. The post-election phase is mainly concerned with the counting of votes and the announcement of election results.

Elements of an electronic voting system

According to (Kho et al., 2022) a generic electronic voting scheme involves the following entities:

- Voter: Individuals who are eligible to vote for candidates.
- Candidate: Nominees seeking to be considered in the election.
- Registrar: Registrars are responsible for authenticating voters.
- Authority: Responsible of conducting the election.
- Auditor: Authorized to verify and review election results.
- Adversary: Malicious individuals attempting to corrupt elections. There are two main types of adversaries, external and internal (Li et al., 2014). External adversaries actively force voters to vote in certain ways, while internal adversaries attempt to breach the system and corrupt voter's privacy and authority.

Homomorphic Encryption

It has been called "The Swiss Army Knife of cryptography" (Halevi, 2017), it is a form of encryption that allows calculations to be performed on encrypted data without first having to decrypt it. An important feature in the process of storing, scrutinizing votes and publishing results.

A cipher system is said to be homomorphic if it is able to perform a particular algebraic operation on an original text, equivalent to another algebraic operation (it may not be the same) on the ciphertext of the original text (Marcolla et al., 2022). Figure 1 shows how it works and then a theoretical breakdown is made according to (Cedillo, 2017).

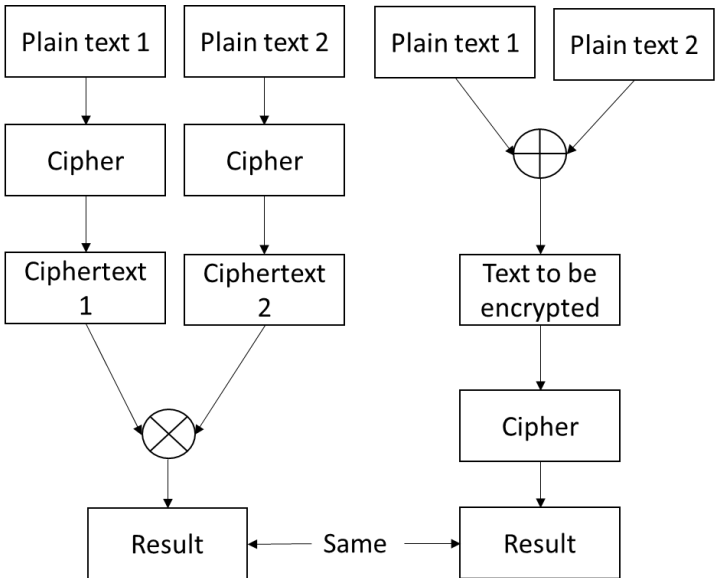
Homomorphic Encryption (HE) is categorized into Fully Homomorphic Encryption (FHE) and Partially Homomorphic Encryption (PHE) methods. FHE supports both addition and multiplication operations on encrypted data. In contrast, PHE refers to a system that allows only

one type of operation, either addition or multiplication. This distinction is discussed in (Ryu et al., 2023).

Gentry's scheme was the first algorithm to demonstrate the feasibility of a fully functional homomorphic encryption system (Gentry, 2009). Proposed by Craig Gentry in 2009, its publication revolutionized the field of cryptography by proving that it is possible to perform arbitrary operations on encrypted data without compromising security

Figure 1

Homomorphic Encryption Scheme



A public-key encryption scheme is homomorphic if for all n and all key pairs (pk, sk) it is possible to define two groups (C, M), such that:

- The plaintext space is M and all ciphertexts obtained from Encpk are elements of C.
- For any m1, m2 ∈ M and c1, c2 ∈ C with m1 = Decsk(c1) and m2 = Decsk(c2) it is satisfied that Decsk(c1 °c2) = m1 + m2, where the group operations are performed on C and M respectively.

There are two operations that can be performed on the elements of the set C and M which are product and addition. The Paillier addition-based algorithm is explained below, and it is currently one of the most widely used in electronic election protocols.

Paillier Cryptographic System (addition-based)

Proposed by Pascal Paillier in 1999 (Paillier, 1999) it is a probabilistic algorithm that uses public key cryptography and is based on the problem of decomposition of number into its prime factors like RSA. Its homomorphic property is defined as follows:

$$Enc_k(m_1) \circ Enc_k(m_2) = [((1+N)^{m_1} \circ r_1^N) \circ ((1+N)^{m_2} \circ r_2^N)\ ] \ mod\ N^2$$

$$= [(1+N)^{m_1+m_2} \circ\ (r_1 \circ r_2)^N] \ mod\ N^2$$

$$= Enc_k(m_1 + m_2\ mod\ (N))$$

Where: N = modulus and r-value = random number

This means that the product of the ciphertexts is equivalent to the sum of the plain texts.

The above homomorphic property can be used by secure electronic voting systems. For example, if n voters are allowed to cast a vote m. Each voter encrypts his choice before submitting his vote. The election authority takes the product of the encrypted votes c and then decrypts the result and obtains the value t, which is the sum of all votes. The election authority then knows the result of the votes without using its private key to decrypt each vote. The role of a random r-value ensures that two equivalent votes will encrypt to the same value with very low and negligible probability, which will guarantee the voter's privacy and confidentiality.

Practical considerations

According to the paper presented by (National Academies of Sciences, Engineering, and Medicine, 2018), e-voting is a cybersecurity problem that requires many factors to consider before it can be implemented in real-world applications. Cybersecurity is an ongoing challenge because adversaries constantly implement new techniques to breach system defenses. E-voting systems connected to the Internet are the most vulnerable to attacks via wireless or

physical access and during data transmission. All e-voting systems, including polling place voting and remote i-voting, are vulnerable to the following attacks.

Denial-of-service (DoS) attacks. The primary objective of DoS attacks is to slow down computer systems and to the extent that it affects the casting of ballots, vote counting and the auditing process.

Malware attacks. Malicious software that can disrupt the vote casting and auditing process and alter or destroy stored ballots.

Malicious individuals or servers break into the system to retrieve sensitive administrator-level data, such as voter credentials.

METHODOLOGY

For the software development, the agile project management framework Scrum is used as a reference and based on some of its artifacts such as Product Backlog, Sprint Backlog and Sprints (Scrum.org, 2024).

Requirements

1. Pre-election

a) Select encryption keys for the Electoral Authority (to encrypt each vote).

b) Select the number of voters v (voter list).

c) Define the number of candidates k (options)

2. Election

a) Voting: votes are generated randomly,

b) Encrypt each vote with Paillier.

3. Post-election

a) The electoral authority calculates the results by multiplying the ciphertext of each vote by taking advantage of the homomorphic property of Paillier and without decrypting each vote.

b) Decrypt the result of the multiplication to obtain the result

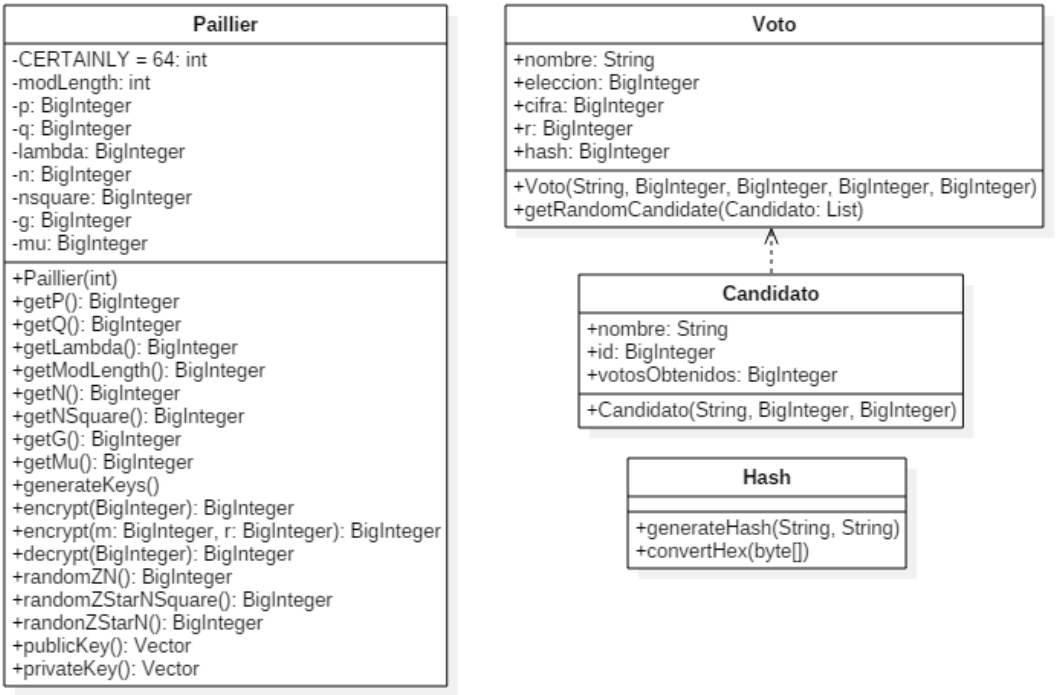
c) Publication of election results in graphs and tables

Development Stage

The application uses the class diagram in Figure 2 for the related cryptographic calculations.

Figure 2

Class diagram used in web application



Interfaces were designed and developed with Java JSP and considering the established requirements.

Testing

Tests were performed for each component. These are based on the execution, review and feedback of the functions previously established in the requirements. Test cases were created, and the results were tabulated obtaining successful results for each one.

RESULTS (TECHNICAL PROPOSAL)

The scheme considered in the present work is based on an algorithm whose challenge is to solve complex mathematical problems or problems of high computational cost, whose strength has been demonstrated.

The Paillier cryptographic system is one of the most widely used cryptographic schemes. It has a wide range of applications, such as banking security systems, in the area related to cloud computing and is widely used in systems for electronic elections (Kho et al., 2022) which is why it has been considered for analysis in this paper.

Paillier homomorphic property is: the product of the ciphertexts is equivalent to the sum of the plaintexts, it is an important feature used to count the ballots in an electronic voting system. When the electoral authority decrypts the result of the products of the ciphertexts, it will get the final result of the election, but it will not know which voter voted for which candidate, ensuring privacy (secrecy).

For the research, the Paillier algorithm has been used, a web application has been developed that acts as an electoral authority and simulates: The generation of the private-public key pair, generate and encrypt random votes, random candidates and publication of results.

Private-public key pair generation

The electoral authority that organizes the election must use a private-public key pair, the public key will be used by the voters to encrypt their votes and the private key for the calculation

of the results to be published, this using its homomorphic property, first obtaining the multiplication of the ciphertexts (votes).

To create a private-public key pair, we first generate two large primes p, q that satisfy mcm (pq, (p-1) (q-1) (q-1)) = 1.

Let n = pq and λ = (p-1)(q-1)/gcd(p-1,q-1) . Then generate a number g such that $g \in \mathbb{Z}_{n^2}^*$ It is ensured that n divides g by computing the existence of:

$$\mu = \left(L\left(g^{\lambda}(mod\ n^2)\right)\right)^{-1}(mod\ n)$$

Where: $L(u) = \frac{u-1}{n}$

As a result, the public key pk is then (n, g), and the private key sk is (λ, μ).

According to Figure 3 the generatekeys() method performs the following tasks:

Figure 3

Code that generates the values for the keys

```
modLegth = keySize;
p = new BigInteger (modLength / 2, CERTAINTY, new Random ());
q = new BigInteger (modLength / 2, CERTAINTY, new Random ());
lambda = (p. subtract (BigInteger.ONE). multiply (q. subtract (BigInteger.ONE))). Divide (subtract
(BigInteger.ONE). gcd(q.subtract(BigInteger.ONE)));
n = p.multiply(q);
nsquare = n.multiply(n);
g = randomZStarNSquare ();
mu = g.modPow(lambda, nsquare). subtract (BigInteger.ONE). divide(n). modInverse(n);
```

Generate random votes and candidates (simulate election)

For the generation of random votes and candidates we will first explain the Paillier encryption algorithm which is as follows:

Given a message $m \in \mathbb{Z}_n$, generate a random number $r \in \mathbb{Z}_n^*$. The encrypted message is: $c = g^m r^n (mod\ n^2)$

For the generation of candidates, it is considered to assign to each candidate a message m that should be placed on the ballot of the election, this m must be calculated according to the number of voters (v) and the number of candidates (k) using base 10 powers to facilitate the computation of the result, using the following code in Figure 4 a different m is obtained for each candidate.

Figure 4

Code that generates k candidates

```
Candidatos = k;
List<Candidato> listaCand = new ArrayList();
for (int i =0; i<candidatos;i++) {
    String nombre = "C"+i;
BigInteger id = BigInteger.TEN.pow(String.valueOf(v).length()*((i+1)-1));
    Candidato c = new Candidato (nombre, id);
    listaCand.add(c);
```

The votes are generated by encrypting a randomly chosen option from the list of candidates and with a random *r* value that makes it impossible for two equivalent votes to have the same ciphertext, code is shown in Figure 5.

Figure 5

Random vote generator routine.

```
for(int i=0; i<votantes;i++) {
   String nombre="V"+i;
   BigInteger eleccion=Voto.getRandomCandidato(listaCand);
   BigInteger r=paillier.randomZStarN();
   BigInteger cifra=paillier.encrypt(eleccion,r);
   String hash=Hash.generarHash(String.valueOf(cifra), Hash.SHA256);
   Voto v = new Voto (nombre, eleccion, cifra, r, hash);
   listaVotos.add(v); }
```

Once the voting has been simulated, we move on to the following step.

Publication of election results

To produce the election results it is necessary to know the decryption algorithm, which is as follows:

Given a ciphertext $c \in \mathbb{Z}_{n^2}^*$, the plaintext message is:

$$m = L\left(c^{\lambda}\ (mod\ n^2)\right)\mu(mod\ n)$$

Then given the homomorphic property we must also calculate the multiplication of the ciphertexts as follows:

$$C_T = \prod_{i=1}^{v} c_i mod\ n^2$$

The code that generates the sum of the figures and the result of the vote counting is shown in Figure 6.

Figure 6

Ciphertext multiplier and result decryption routine.

```
BigInteger sumCifra=BigInteger.ONE;
      for (Voto v: listaVotos) {
         sumCifra = sumCifra.multiply(v.getCifra());}
sumCifra = sumCifra.mod(paillier.getN(). pow(2));
int longitud = String.valueOf(votantes). length();
conteo = String.format("%0"+candidatos*longitud+"d", paillier.decrypt(sumCifra));
String [] votosPorCandidato = conteo.split("(? <=\\G. {" + longitud + "})");
out.println("Resultado Conteo (cifras multiplicadas): "+ conteo);
```

Zero Knowledge Proof

These proofs are part of the audit that an election must undergo in order to ensure its transparency and validity.

Zero Knowledge Proof (ZKP) allows an individual to convince another individual that he has certain information without revealing anything about the content of the information. In

DOI: https://doi.org/10.71112/f449y553

electronic voting schemes there are two secrets to protect primarily, one is the vote, and the other is the secret encryption key (Garimella & Conway, 2024).

At least 2 tests should be performed: one to verify that the encrypted vote is valid, and another for the electoral authority to prove that the final decryption was performed correctly. The description of the tests can be found in (Cedillo, 2017).

In the first one, when voters send their votes to be counted, the electoral authority must be able to prove that the cipher of their vote is one of {1, N, N2,..., Nm-1}. To do this, a ZKP is performed between a voter and the election authority.

In the second one, it is desired to know if the final result T (plaintext of the voting result) obtained from C_T is correct. In this case, the objective is to prove to any entity that the result has been correctly decrypted, but without revealing the private key that was used.

Graph and tabulated example of election results.

Below is an example of the resulting simulation with the application developed in Java with JSP.

The graph is shown in Figure 7, while Table 1 shows the votes in detail.

Input data used: keySize = 2048, v =100, k = 4

Figure 7

Graph showing the simulation result.

Table 1

Published election result (figures shortened for presentation purposes to 10/100 voters and 10/600 digits approx.).

Vote	R	Candidate	Ciphertext	Tracking code
V0	1898263655	1	1502743253	78be8b1484
V1	1654883641	1000000	1378275499	83e7115dcd
V2	2173069031	1000000	1256832105	77dca46bb0
V3	8643859050	1	1551794613	583c203c76
V4	1809870043	1000	1890018695	d927e56aa6
V5	1577341515	1000	2261629092	4c081de864
V6	9305778105	1	1443214753	9154aa5f9b
V7	9706827489	1000000000	1658896460	aa27ea4f11
V8	6873537249	100	3174567650	0ec2545611
V9	1322381662	100	8692403207	5eab40716f

This example is established for demonstrative purposes because the election should not be revealed in clear, nor the variable *r*, since this compromises the security, privacy and secrecy of the voters, in addition a tracking code is created to perform audit tests between the voter and the electoral authority.

DISCUSSION

The results obtained in the simulation confirm the feasibility of using Paillier's homomorphic encryption in an electronic voting scheme, particularly due to its capacity to preserve vote privacy while enabling tallying without decrypting individual ballots. This approach addresses one of the main challenges of online voting systems: ensuring both the integrity of the election outcome and the anonymity of voters.

The use of the additive homomorphic property demonstrates that it is possible to sum votes through the multiplication of ciphertexts, minimizing the handling of plaintext data and consequently reducing the risk of sensitive information exposure. This result is consistent with

(Cedillo, 2017), who emphasized that homomorphic encryption allows transparency in the process without compromising individual confidentiality.

Compared with other cryptographic schemes used in international experiences, such as ElGamal or AES in the Australian case (Halderman & Teague, 2015), the Paillier algorithm offers advantages by avoiding the need to decrypt votes for processing, thus mitigating risks of linking identities to ballots. Furthermore, the practical demonstration through a web-based application supports the theoretical assertions by (Halevi, 2017) regarding the potential of homomorphic encryption as a versatile and secure tool for high-sensitivity systems, including electronic elections.

Nevertheless, there are still significant limitations for real world adoption. A secure electronic voting implementation depends not only on the cryptographic algorithm but also on the underlying technological infrastructure, key management, and auditing mechanisms. As observed in the cases of Estonia, Switzerland, and Russia (Gaudry & Golovnev, 2020; Springall et al., 2014; Zetter, 2019), even systems based on robust algorithms can be vulnerable due to design flaws, verification gaps, or misconfigurations.

Additionally, the inclusion of Zero-Knowledge Proofs (ZKP) is presented as an essential component to ensure transparency and verifiability, both during vote casting and tallying. These proofs, already implemented in Swiss systems and discussed by (Haines et al., 2020), make it possible to guarantee that the process functions correctly without revealing sensitive information.

From a practical standpoint, the simulation demonstrated that homomorphic vote counting can be efficiently performed even with 2048-bit keys, indicating its feasibility for small and medium-scale electoral processes. However, scalability and resilience against denial-of-service attacks or malicious software manipulation remain relevant challenges (National Academies of Sciences, Engineering, and Medicine, 2018).

In summary, this study shows that combining Paillier's homomorphic encryption with cryptographic auditing mechanisms, such as Zero-Knowledge Proofs, provides a solid foundation for trustworthy electronic voting systems. Nevertheless, the overall security of such systems will depend on the correct implementation of software, network infrastructure, independent auditing, and voter awareness of cybersecurity principles. Future research could extend this proposal by integrating post-quantum cryptography and blockchain-based architectures to enhance resistance against emerging technological threats.

CONCLUSIONS

The Paillier algorithm correctly implemented in electronic voting is infeasible because it has passed the test of time, so cryptographic vulnerabilities must be discarded.

The efficiency of the vote counting process and publication of election results are increased in terms of efficiency compared to manual counting mechanisms at the ballot box.

The integrity and privacy of the counted votes is ensured under the correct application of.

Paillier's cryptographic system for vote storage.

The audit of the elections as part of the transparency and validity can be ensured by the properties of the cryptographic system, both between the voter and the electoral authority, as well as the result published through a Zero-Knowledge Proof (ZKP).

E-voting is a cybersecurity problem that requires many factors to be considered before it can be implemented in real-world applications. Cybersecurity is an ongoing challenge because adversaries are constantly implementing new techniques to breach system defenses.

The current work could evolve into a Software Development Kit (SDK) for straightforward implementation of e-voting systems, from creation to verification and publication of election results.

Recommendations

In the development of electronic voting applications, security must be implemented as a transversal axis to secure networks, systems and IT infrastructure, which are the edges from which vulnerabilities can be induced.

Apply fingerprinting to be able to audit the integrity of certain files such as: the electoral roll, the source code of the voting application and/or the ballot.

Perform ZKPs, one to verify that the encrypted vote is valid, and another for the electoral authority to demonstrate that the final decryption was performed correctly. The description of the proofs can be found in (Cedillo, 2017).

Use in combination with mixnet such as TOR to guarantee anonymity and make it impossible to trace the origin (user) and destination (server) of the vote.

Quantum computing can break any current cryptographic scheme, including Blockchain (Mamatha et al., 2024), but also not ready yet, so work is already underway on post-quantum e-voting schemes. Post-quantum cryptography in e-voting schemes is a new research direction and few have implemented it in e-voting.

Declaración de conflicto de interés

El autor declara no tener ningún conflicto de interés relacionado con esta investigación.

Declaración de contribución a la autoría

Álvaro Hernán Zavala Ruballo: conceptualización, curación de datos, análisis formal, adquisición de fondos, investigación, metodología, administración del proyecto, recursos, software, supervisión, validación, visualización, redacción del borrador original, revisión y edición de la redacción

Declaración de uso de inteligencia artificial

El autor no utilizó inteligencia artificial en ninguna parte del manuscrito.

REFERENCES

Abby Abazorius. (2020, February 13). MIT researchers identify security vulnerabilities in voting app. MIT News | Massachusetts Institute of Technology. https://news.mit.edu/2020/voting-voatz-app-hack-issues-0213

Cedillo, M. (2017). Sistema de Votación por Internet FIDELIS. CINVESTAV.

Chaum, D. L. (1981). Untraceable electronic mail, return addresses, and digital pseudonyms. Commun. ACM, 24(2), Article 2. https://doi.org/10.1145/358549.358563

Collins English Dictionary. (2024). E-VOTING definition and meaning | Collins English Dictionary. https://www.collinsdictionary.com/dictionary/english/e-voting

Drummond, M. (2024). An Introduction to Cryptography. IDPro Body of Knowledge, 1(13). https://doi.org/10.55621/idpro.102

Garimella, K. K., & Conway, D. (2024). Zero-Knowledge Proofs and Privacy: A Technical Look at Privacy. In M. C. Lacity & L. Coon (Eds.), Human Privacy in Virtual and Physical Worlds: Multidisciplinary Perspectives (pp. 157–179). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-51063-2_8

Gaudry, P., & Golovnev, A. (2020). Breaking the Encryption Scheme of the Moscow Internet Voting System. In J. Bonneau & N. Heninger (Eds.), Financial Cryptography and Data Security (Vol. 12059, pp. 32–49). Springer International Publishing. https://doi.org/10.1007/978-3-030-51280-4_3

Gentry, C. (2009). A fully homomorphic encryption scheme. https://api.semanticscholar.org/CorpusID:53903759

Haines, T., Lewis, S. J., Pereira, O., & Teague, V. (2020). How not to prove your election outcome. 2020 IEEE Symposium on Security and Privacy (SP), 644–660. https://doi.org/10.1109/SP40000.2020.00048

Halderman, J. A., & Teague, V. (2015). The New South Wales iVote System: Security Failures and Verification Flaws in a Live Online Election (No. arXiv:1504.05646). arXiv. https://doi.org/10.48550/arXiv.1504.05646

Halevi, S. (2017). Homomorphic Encryption. In Y. Lindell (Ed.), Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich (pp. 219–276). Springer International Publishing. https://doi.org/10.1007/978-3-319-57048-8_5

Kartit, Z., Azougaghe, A., Kamal Idrissi, H., El Marraki, M., Hedabou, M., Belkasmi, M., & Kartit, A. (2016). Applying Encryption Algorithm for Data Security in Cloud Storage. In E. Sabir, H. Medromi, & M. Sadik (Eds.), Advances in Ubiquitous Networking (pp. 141–154). Springer Nature Singapore.

Kho, Y.-X., Heng, S.-H., & Chin, J.-J. (2022). A Review of Cryptographic Electronic Voting. Symmetry, 14(5), Article 5. https://doi.org/10.3390/sym14050858

Li, H., Kankanala, A. R., & Zou, X. (2014). A taxonomy and comparison of remote voting schemes. 2014 23rd International Conference on Computer Communication and Networks (ICCCN), 1–8. https://doi.org/10.1109/ICCCN.2014.6911807

Mamatha, G. S., Dimri, N., & Sinha, R. (2024). Post-Quantum Cryptography: Securing Digital Communication in the Quantum Era (No. arXiv:2403.11741). arXiv. https://doi.org/10.48550/arXiv.2403.11741

Marcolla, C., Sucasas, V., Manzano, M., Bassoli, R., Fitzek, F. H. P., & Aaraj, N. (2022). Survey on Fully Homomorphic Encryption, Theory, and Applications (No. 2022/1602). Cryptology ePrint Archive. https://doi.org/10.1109/JPROC.2022.3205665

Menezes, A., Oorschot, P. C. van, & Vanstone, S. A. (1997). Handbook of Applied Cryptography. CRC Press.

National Academies of Sciences, Engineering, and Medicine. (2018). Securing the Vote: Protecting American Democracy. Washington. 2018. Securing the Vote: Protecting

American Democracy. Washington, DC: The National Academies Press. Https://doi.org/10.17226/25120. The National Academies Press. https://nap.nationalacademies.org/catalog/25120/securing-the-vote-protecting-american-democracy

NIST. (2015). Secure Hash Standard (SHS) (No. Federal Information Processing Standard (FIPS) 180-4). U.S. Department of Commerce. https://doi.org/10.6028/NIST.FIPS.180-4

NIST. (2023a). Advanced Encryption Standard (AES) (No. NIST FIPS 197-upd1; p. NIST FIPS 197-upd1). National Institute of Standards and Technology (U.S.). https://doi.org/10.6028/NIST.FIPS.197-upd1

NIST. (2023b). Digital Signature Standard (DSS). National Institute of Standards and Technology (U.S.). https://doi.org/10.6028/NIST.FIPS.186-5

OASIS. (2011). Election Markup Language (EML) Specification Version 7.0. https://docs.oasis-open.org/election/eml/v7.0/eml-v7.0.html

Paillier, P. (1999). Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In J. Stern (Ed.), Advances in Cryptology—EUROCRYPT '99 (Vol. 1592, pp. 223–238). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-48910-X_16

Peng, K. (2011). An efficient shuffling based eVoting scheme. Journal of Systems and Software, 84(6), 906–922. https://doi.org/10.1016/j.jss.2011.01.001

Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM, 21(2), Article 2. https://doi.org/10.1145/359340.359342

Ryu, J., Kim, K., & Won, D. (2023). A Study on Partially Homomorphic Encryption. 2023 17th International Conference on Ubiquitous Information Management and Communication (IMCOM), 1–4. https://doi.org/10.1109/IMCOM56909.2023.10035630

Scrum.org. (2024). Scrum. https://www.scrum.org/learning-series/what-is-scrum/

Shirey, R. W. (2007). Internet Security Glossary, Version 2 (Request for Comments No. RFC 4949; Issue RFC 4949). Internet Engineering Task Force. https://doi.org/10.17487/RFC4949

Springall, D., Finkenauer, T., Durumeric, Z., Kitcat, J., Hursti, H., MacAlpine, M., & Halderman, J. A. (2014). Security Analysis of the Estonian Internet Voting System. Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, 703–715. https://doi.org/10.1145/2660267.2660315

Zetter, K. (2019, March 12). Researchers Find Critical Backdoor in Swiss Online Voting System. Vice. https://www.vice.com/en/article/zmakk3/researchers-find-critical-backdoor-in-swiss-online-voting-system