UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

COLEGIO DE CIENCIAS E INGENIERÍA

Mathematical Principles and Applications of Blockchain Technology in E-Voting Systems

Iván Emilio Guerrón Calero

Matemáticas

Trabajo de fin de carrera presentado como requisito
para la obtención del título de
Matemático

Quito, 12 de diciembre de 2024

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

COLEGIO DE CIENCIAS E INGENIERÍA

HOJA DE CALIFICACIÓN

DE TRABAJO DE FIN DE CARRERA

Mathematical Principles and Applications of Blockchain Technology in E-Voting Systems

Iván Emilio Guerrón Calero

Nombre del profesor, Título académico **Julio Ibarra, Ph.D**

Quito, 12 de diciembre de 2024

@DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en la Ley Orgánica de Educación Superior del Ecuador.

Nombres y Apellidos:	Iván Emilio Guerrón Calero
Código:	00334838
Cédula de Identidad:	1723269526
Lugar y Fecha:	Quito, 12 de diciembre de 2024

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en http://bit.ly/COPETheses.

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on http://bit.ly/COPETheses.

RESUMEN

El uso de la tecnología blockchain en los sistemas de voto electrónico tiene un gran potencial para aportar soluciones a los problemas de seguridad, transparencia e integridad de los actuales sistemas electorales. Este artículo analiza los antecedentes matemáticos de la tecnología blockchain, con especial referencia a la criptografía de curva elíptica y las funciones hash criptográficas que garantizan la solidez del sistema de votación. Esta investigación hace hincapié en la capacidad de blockchain para transformar los procesos electorales, de modo que los procesos democráticos puedan asegurarse, hacerse transparentes y accesibles gracias a los avances tecnológicos.

Palabras clave: Blockchain, bloques, funciones hash criptográficas, curvas elípticas, algoritmo de firma digital de curva elíptica, voto electrónico, raíz de Merkle, SHA-256.

ABSTRACT

The usage of blockchain technology in e-voting systems holds a lot of potential in providing solutions to the problems of security, transparency and integrity in the current electoral systems. This paper discusses the mathematical background of the blockchain technology, with particular reference to elliptic curve cryptography and cryptographic hash functions that guarantee the solidity of the voting system. This research emphasizes on the ability of blockchain to transform the electoral processes so that democratic processes can be secured, made transparent, and made accessible through technological advancement.

Keywords: Blockchain, blocks, cryptographic hash functions, elliptic curves, Elliptic Curve Digital Signature Algorithm, e-voting, Merkle root, SHA-256.

TABLE OF CONTENTS

INDEX OF TABLES

INDEX OF FIGURES

INTRODUCTION

Blockchain for Elections

In the recent past, the idea of implementing blockchain in the elections has received a lot of support because of the ability that can transform the voting system. Some of the problems that are associated with the conventional systems of voting either on paper or electronic systems include security, transparency and trust. Some of these concerns are: manipulation of votes, cases of people voting multiple times, and the general manner in which the whole process is conducted. Hence, people living in democratic societies require more protection and transparency in the electoral processes, the importance of such solutions is evident.

Blockchain technology is an innovative solution to these challenges because it provides a decentralized and transparent system that can be used to store and authenticate votes in a secure manner, minimizing the possibility of vote manipulation. Every vote is considered as a transaction on the blockchain and it is very difficult to tamper with the information after it has been entered into the ledger.

Trust and transparency should be the main focus of any voting system. Due to the decentralised nature of blockchain, all the participants in the network have the full history of votes and hence the voting process is transparent and can be audited. This level of transparency assists in building trust among the voters since they are able to check if their vote has been recorded and counted in the right manner. Moreover, the employment of cryptographic approaches ensures that the votes cast are both safe and will not be traced back to the individuals who cast them hence privacy is well protected while at the same time the elections are credible.

The application of blockchain technology can be considered as promising and effective

solution to the issues which are related to the voting systems. We can use the properties of blockchain that are decentralised, immutable and transparent to develop a system by which people can vote and be certain that every vote is captured and the electoral process is not rigged.

Motivation

The idea behind e-voting, blockchain-based systems have been inspired by emerging danger against the security, openness and fairness of elections in the present world. Electoral activities in most societies are characterized by accusations of rigging, vote manipulation, and the like, lacking in democracy. Such problems do not only call into question the validity of results but also work towards demolition of people's confidence in democracy project. While bearing in mind that technology is a tool, there has been a rise in ways of manipulating and meddling with the electoral processes to achieve a desired goal hence making it extremely difficult to have a free and fair electoral process.

The need to use blockchain technology to support e-voting system is also fueled by the need to provide power to voters and protect their power. In many countries voter manipulation and people's exclusion from the voting process is a critical problem that prevents a significant number of people from going to the polls. An example of this is a voting system that operates through the blockchain: such a system can be convenient for voters, as they can vote without leaving their homes and without having to go to the polls. This level of accessibility can give a boost to the turnout and make sure that every voter that is legally allowed to vote, will do so.

Furthermore, there are other advantages of implementing the e-voting system using the blockchain technology; it can also cut down the expenses and time required in the electoral

process. Paper based voting systems are usually very costly to administer and maintain since they involve printing of ballots, setting up of voting stations, and manual tallying of votes. Some of the costs involved include; Through the use of digital voting and the application of blockchain technology most of these costs can easily be done away with or at least minimized in the voting system.

In conclusion, the need to adopt an e-voting system that will be based on blockchain technology is to promote a better electoral process that is more secure, transparent and more accessible. Thus, with the help of the features offered by the blockchain, it is possible to eliminate the problems of the classical voting system and create a platform that will allow the voter to protect his rights and ensure the transparency of the elections at the present stage.

The History of blockchain

Back in 1991, two American researchers, a cryptographer and computer scientist, and a physicist, named Wakefield Scott Stornetta and Stuart Haber, respectively, presented a paper discussing the importance of time-stamping documents or data [1]. They realized that with the increasing usage of documents (texts, audio, photos, and videos), a new problem came along; when were these documents last tampered with? Hence, they proposed a cryptographic approach to secure the integrity of the documents, but without the need to rely on any centralized TTS (Time-stamp Service). The idea behind this approach was to distribute the trust between the users of such service, creating a ”chain of time-stamps” or hashes, in which each time-stamp not only contained the hash of the current document, but of the previous one as well.

Linked time-stamping, described by Haber and Stornetta, would later develop into the foundation of blockchain. Their work brought into focus the need to have secure, tamper-proof,

and auditable digital data. This concept, however, deserved more thoughts and ideas to make a breakthrough and cover the other aspects and possibilities of such a system.

In the following years, several developments were made in cryptographic methods and distributed networks. One of the main innovations was the appearance of Proof-of-Work (PoW) algorithms. In 1993, Moni Naor and Cynthia Dwork introduced the idea behind PoW as an approach to fight against spam emails [2]. PoW, at its core, is designed to require a certain amount of computation to be completed before a service can be used, making resource usage come at a cost for potential abusers.

This was expanded upon by Adam Back in 1997 with the re-invention of hashcash as a cost-function for the creation of tokens that could be used as PoWs [3]. Hashcash implies that the sender calculates a hash value that should meet specific conditions as a sign that computation was done. In the junk mails context, this mechanism made sure that spamming became economically unprofitable to undertake because of the many computational resources required.

All these concepts allowed for one of the most technologically relevant white papers in history to be written. In 2008, Satoshi Nakamoto (pseudonym for a person or group) published ”Bitcoin: A Peer-to-Peer Electronic Cash System” [4]. Nakamoto, through this white paper, introduced the world to a decentralized digital currency that would use the block chain technology to ensure the validity of a transaction and to ensure that there was trust without a central control.

Nakamoto’s idea was to combine PoW with the network of nodes to make the system more secure for the record of the transaction. In Bitcoin network, miners employ their computational facility to solve very complex mathematical computations known as Proof-of-Work (PoW) to authenticate transactions and add blocks in the blockchain. This process is indispens-

able for the protection of the network; plus it encourages participants to attain fresh satoshis (fractions of bitcoins).

The Bitcoin blockchain works in the same way as other distributed ledgers, in which each block has a list, a timestamp, and the previous block's hash. This structure makes it virtually impossible for anyone to attempt to change a transaction because, if he or she did so, the PoW for that block and all the following blocks would have to be redone.

After the emergence of the Bitcoin, the idea of blockchain attracted a lot of attention and led to the creation of many other cryptocurrencies and decentralized applications. The Ethereum that was launched by Vitalik Buterin in the year 2015, enhanced the applications of blockchain technology through the addition of the smart contracts [5]. These digital contracts, which terms are coded into the system, are automatically executed without external or third-party intervention.

Ethereum's innovation was to show that blockchain could be used for more than just decentralized digital currencies. That proved that blockchain can be utilized in establishing the decentralized autonomous organizations or DAOs, voting systems, supply chain management (SCM), and many others. Smart contracts emerged as the next innovation in the blockchain world, which allowed for programmable and self-executing transactions.

At the same time, other consensus algorithms were considered to complement the drawbacks of PoW, including high energy consumption. Proof-of-Stake (PoS) appeared as the second proposed, in 2012 by Sunny King and Scott Nadal in the Peercoin cryptocurrency [6]. PoS involves the selection of validators in proportion to the amount of coins they are willing to lock and 'stake' on the network. This approach cuts energy use and is scalable but at the same time, is secure and decentralized.

Since then, many blockchain projects have incorporated, or have suggested some forms of PoS, such as the Ethereum network's current upgrade to Ethereum 2.0, which plans to upgrade its consensus algorithm to PoS in order to accommodate more transactions per second and lessen the carbon footprint. Other consensus algorithms include Proof-of-Authority, Delegated Proof-of-Stake, and Byzantine Fault Tolerance with each providing a varying level of security, speed, and decentralization.

MATHEMATICS BEHIND BLOCKCHAIN

Elliptic Curves

Definition 1. *A binary operation on a set G is a function $G \times G \longrightarrow G$ that assigns to each pair $(a,b) \in G \times G$ a unique element $a * b \in G$, also known as the composition of a and b [7].*

Definition 2. *[7] A group $(G,*)$ is a set G together with a binary operation $(*)$ which form an algebraic structure, and satisfies the following properties:*

1. ***Associativity**: $\forall\ a,b,c \in G,\ (a*b)*c = a*(b*c)$.*
2. ***Identity Element**: $\exists\ e \in G$, such that $\forall\ a \in G,\ a*e = e*a = a$.*
3. ***Inverse Element**: $\forall\ a \in G,\ \exists\ a^{-1} \in G$, such that $a*a^{-1} = a^{-1}*a = e$.*

Definition 3. *[7] G is called a commutative or abelian group if in addition to the three main group properties, it satisfies the following:*

- $\forall\ a,b \in G, b*a = a*b$.

Definition 4. *[7] A field $\mathbb{F}$ is a non-empty set, where two binary operations are defined, known as addition and multiplication, such that for any $a,b,c \in \mathbb{F}$:*

1. *$a+(b+c) = (a+b)+c$ and $a(bc) = (ab)c$.*
2. *$\exists\ 0 \in \mathbb{F}$, such that $0+a = a+0 = a$.*
3. *$\exists\ -a \in \mathbb{F}$, such that $(-a)+a = a+(-a) = 0$.*
4. *$a+b = b+a$.*

5. $ab = ba$

6. $(b+c)a = ba+ca$ *and* $a(b+c) = ab+ac$.

7. $\exists\ 1 \in \mathbb{F}$, *such that* $1 \neq 0$ *and* $1a = a1 = a$ *for any* $a \in \mathbb{F}$.

8. *For* $a \in \mathbb{F}$ *and if* $a \neq 0$, *then* $\exists\ a^{-1} \in \mathbb{F}$, *such that* $a^{-1}a = aa^{-1} = 1$.

Definition 5. *A field* $\mathbb{F}$ *is called a* finite field *if it has a finite number of elements. The number of elements in a finite field is called the* order *of the field.*

Definition 6. *A point on a curve* C *defined by* $F(x,y) = 0$ *is called* singular *if the partial derivatives of the curve's equation vanish at that point. If a curve has no singular points, it is called* non-singular *[8].*

Definition 7. *If* E *is a curve defined by the equation* $y^2 = x^3 + Ax + B$ *where* $A, B \in \mathbb{F}$, *it is called an* elliptic curve *[8].*

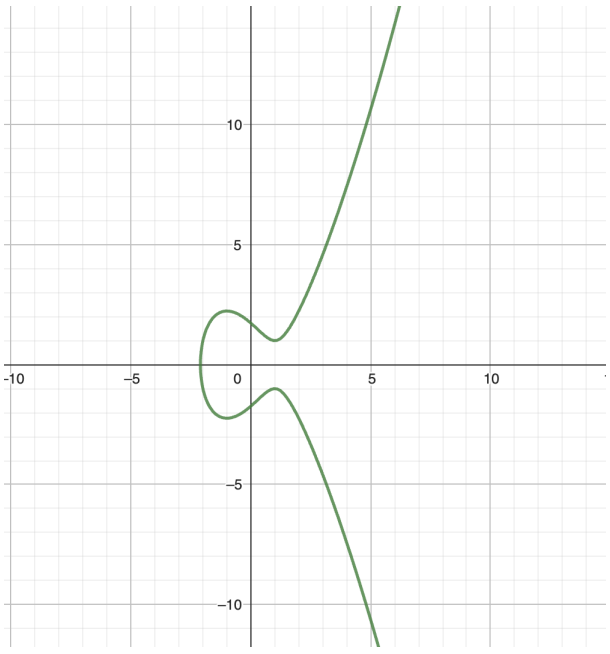

Figure 1: Example of an elliptic curve $y^2 = x^3 - 3x + 3$

Proposition 1. *An elliptic curve* E *over a field* $\mathbb{F}$ *is non-singular if and only if the discriminant* $\Delta = -(4A^3 + 27B^2) \neq 0$ *[8].*

Proof. Let's consider a curve defined by the equation $F(x,y) = 0$. This curve is said to be *singular* at a given point (x_0, y_0) if the partial derivatives of F vanish simultaneously at that given point (by Definition 5). That is:

$$\frac{\partial F}{\partial x}(x_0, y_0) = 0 \quad \text{and} \quad \frac{\partial F}{\partial y}(x_0, y_0) = 0.$$

As for the particular case where the elliptic curve is defined by:

$$y^2 = p(x),$$

with $p(x)$ being a polynomial of x. Therefore, we are left with the function:

$$F(x,y) = y^2 - p(x).$$

Then, the partial derivatives of the curve given by F with respect to x and y are:

$$\frac{\partial F}{\partial x} = -p'(x), \quad \frac{\partial F}{\partial y} = 2y.$$

For us to call the point (x_0, y_0) singular, we require:

$$2y_0 = 0 \quad \text{and} \quad -p'(x_0) = 0.$$

This implies $p'(x_0) = 0$ and $y_0 = 0$, and so $p(x_0) = 0$. Therefore, x_0 is at least a *double root* of $p(x)$, meaning that $(x - x_0)^2$ divides $p(x)$.

Now, let's determine under what conditions the following polynomial

$$p(x) = x^3 + Ax + B$$

has root multiplicity greater than 1. For this, let the roots of $p(x)$ be denoted by u, v, and w. Then, we can see $p(x)$ as:

$$\begin{aligned} p(x) &= (x-u)(x-v)(x-w), \\ x^3 + Ax + B &= (x-u)(x-v)(x-w) \\ &= x^3 - (u+v+w)x^2 + (uv+vw+wu)x - uvw, \end{aligned}$$

obtaining:

$$u+v+w = 0, \quad uv+vw+wu = A, \quad uvw = -B.$$

The discriminant of this cubic polynomial is given by the following expression:

$$\Delta = (u-v)^2(u-w)^2(v-w)^2.$$

The discriminant Δ is important cause it has two indispensable properties:

- It vanishes if and only if two or more of the roots coincide, indicating a multiple root.
- It is a symmetric polinomial. [9]

Given that $u+v+w = 0$, let's substitute $w = -u-v$ and express Δ as:

$$\begin{aligned}
\Delta &= (u-v)^2(2u+v)^2(2v+u)^2 \\
&= 4u^6 + 12u^5v - 3u^4v^2 - 26u^3v^3 - 3u^2v^4 + 12uv^5 + 4v^6 \\
&= -4(uv + u(-u-v) + v(-u-v))^3 - 27(-uv(-u-v))^2 \\
&= -4(uv+uw+vw)^3 - 27(-uvw)^2 \\
&= -(4A^3 + 27B^2)
\end{aligned}$$

Thus, the condition $4A^3 + 27B^2 = 0$ expresses the scenario where two or more of the roots of the cubic polynomial $p(x)$ coincide, meaning the curve $y^2 = p(x)$ is singular.

Therefore, the elliptic curve is non-singular if and only if $\Delta \neq 0$ □

Example 1. *Consider the elliptic curve E defined by $y^2 = x^3 - 2x + 1$ over $\mathbb{Q}$. This curve is non-singular since $\Delta = -4(-2)^3 - 27 \cdot 1^2 = 32 - 27 = 5 \neq 0$. Figure 2 shows this curve and illustrates three different roots on it.*

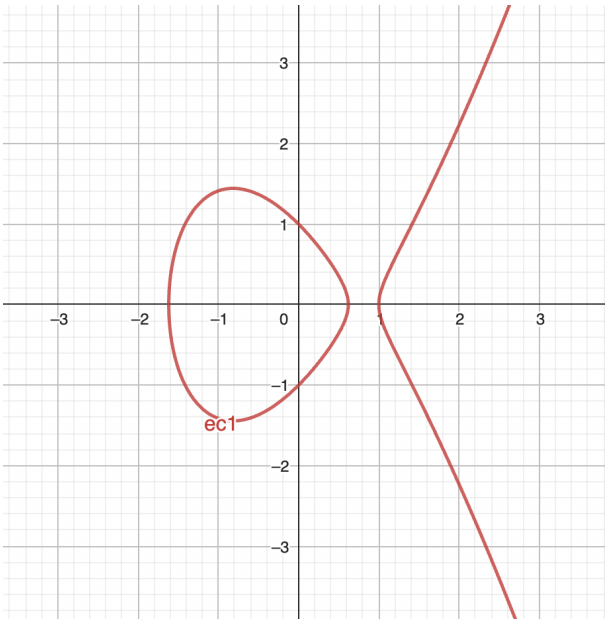

Figure 2: Example of the elliptic curve $y^2 = x^3 - 2x + 1$ over $\mathbb{Q}$

The Group Law

Definition 8. *Let E be an elliptic curve, and let it be defined by the following equation: $y^2 = x^3 + Ax + B$. Let's consider $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ to be points on E, with P_1 and P_2 different from infinity. Then define the sum of P_1 and P_2, denoted $P_1 + P_2 = P_3 = (x_3, y_3)$, as follows:*

1. *If $x_1 \neq x_2$, then*

$$x_3 = m^2 - x_1 - x_2, \quad y_3 = m(x_1 - x_3) - y_1,$$

where

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

Refer to Figure 3 for a visual representation of this case.

2. *If $x_1 = x_2$ but $y_1 \neq y_2$, then $P_1 + P_2 = \infty$.*

3. *If $P_1 = P_2$ and $y_1 \neq 0$, then*

$$x_3 = m^2 - 2x_1, \quad y_3 = m(x_1 - x_3) - y_1,$$

where

$$m = \frac{3x_1^2 + A}{2y_1}.$$

Refer to Figure 4 for a visual representation of this case.

4. *If $P_1 = P_2$ and $y_1 = 0$, then $P_1 + P_2 = \infty$.*

5. *For any P on E, $P + \infty = P$ [8].*

Note that the identity element of the group is the point at infinity. Let's call it O for convenience.

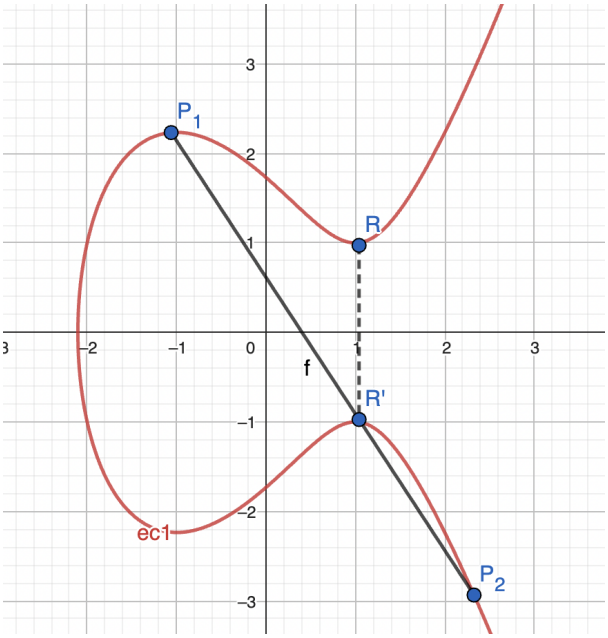

Figure 3: Group law case 1: Addition of two distinct points, $R = P_1 + P_2$.

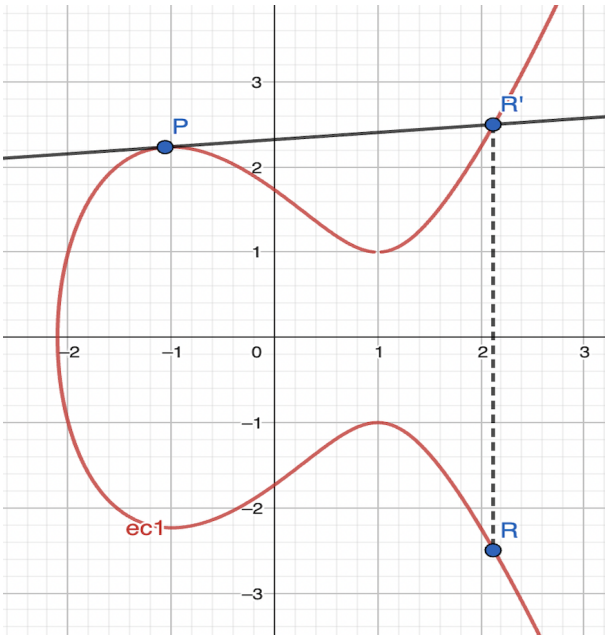

Figure 4: Group law case 3: Doubling a point $R = P + P$ on the curve.

Theorem 1. *If E is an elliptic curve, then $E(\mathbb{F})$ forms a commutative or abelian group under a well-defined addition operation (described above) [8].*

Proof. The addition operation on the points of an elliptic curve is defined geometrically using the chord-tangent method. The proof that $E(\mathbb{F})$ is a group is straight-forward. Furthermore, since for any $P_1, P_2 \in E$, the line that passes through P_1 and P_2 is the same line that passes through P_2 and P_1, it is obvious that it satisfies commutativity. □

Elliptic curves are considered of great relevance in cryptography and number theory thanks to their structure and properties. This relies on the fact that they provide a secure and efficient foundation for cryptographic protocols, such as elliptic curve cryptography (ECC), which extends solid security with relatively small key sizes. This makes them particularly useful for digital signatures, key exchange mechanisms and encryption in environments with limited computational resources [10]. Additionally, the properties of these curves over finite fields leads to complex mathematical problems, like the elliptic curve discrete logarithm problem, which justifies the security of these systems.

Elliptic Curve Cryptography

ECC or Elliptic Curve Cryptography is an asymmetric-key (public-key) cryptographic approach used to encrypt data, authenticate users, and generate and verify digital signatures [10]. Elliptic curves over finite fields are the foundation of its algebraic structure. Since public-key generation in ECC leverages the algebraic properties and characteristics of elliptic curves with the group law, it is widely used amongst security systems (other methods propose to manipulate massive prime numbers to generate secure public-keys).

Definition 9. *The unique finite field of $q = p^n$ elements is called finite field of order q ($\mathbb{F}_q$) or Galois Field of order p^n ($\mathbb{GF}(p^n)$).*

Remark 1. *To prove the existence and uniqueness of this finite field, we would need to introduce*

new concepts and theorems, which are not that relevant for the purposes of this paper and are quite extensive. For the interested reader, refer to [7].

Definition 10. *An elliptic curve over a finite field $\mathbb{F}_q$ is the set of points (x,y), where $x,y \in \mathbb{F}_q$, that satisfy $y^2 \equiv x^3 + Ax + B \pmod{q}$ with $A,B \in \mathbb{F}_q$ and $\Delta \neq 0$ [8].*

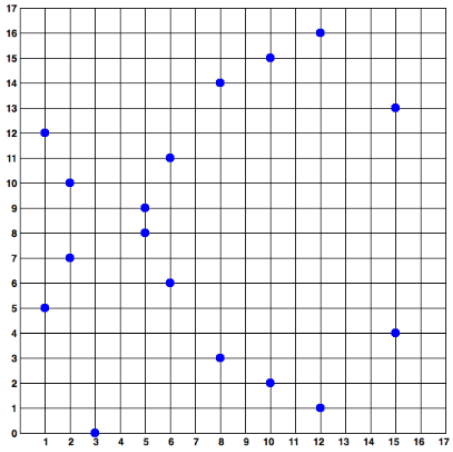

Figure 5: An elliptic curve over a finite field $\mathbb{F}_q$

Definition 11. *Let $E(\mathbb{F}_q)$ be an elliptic curve and let a,b be points on $E(\mathbb{F}_q)$. The Elliptic Curve Discrete Logarithm Problem (ECDLP) describes the search for an integer k such that $ka = b$ [8].*

Remark 2. *The security of ECC relies on the difficulty of the ECDLP.*

Definition 12. *The* Elliptic Curve Diffie-Hellman (ECDH) *key exchange is a cryptographic protocol that allows two parties to establish a shared secret over an insecure channel using elliptic curves [10]. Figure 6 gives us a general idea of how ECDH works.*

Definition 13. *The* Elliptic Curve Digital Signature Algorithm (ECDSA) *is a digital signature scheme based on elliptic curves that provides authentication and data integrity [8]. For more detail on ECDSA, refer to Algorithms 1 and 2.*

Definition 14. *[10] Let $D = (q, FR, a, b, P, n)$ be the domain parameters, composed by:*

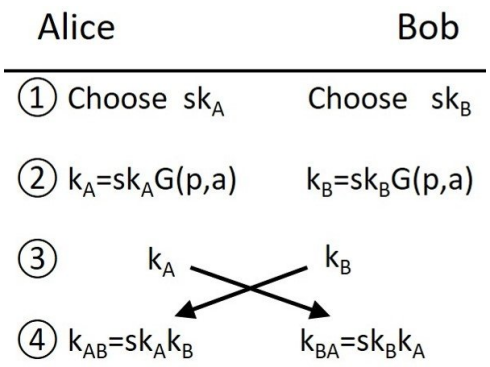

Figure 6: Elliptic Curve Diffie-Hellman (ECDH) key exchange [11]

- *Field order q.*
- *Field representation FR of the elements of* $\mathbb{F}_q$.
- $a,b \in \mathbb{F}_q$ *that define the ellipse curve E over* $\mathbb{F}_q$ *(i.e for* $y^2 = x^3 + ax + b$*).*
- *A finite point* $P = (x_p, y_p) \in E(\mathbb{F}_q)$*, where* $x_p, y_p \in \mathbb{F}_q$ *(P is called base point and possesses prime order).*
- *The order n of P. This means that n is the smallest positive integer such that* $nP = O$.

Algorithm 1 ECDSA: Generation

Require: Domain $D = (q, FR, a, b, P, n)$, signer's private key d_A, hash function $H(\cdot)$, message m to be signed.
Ensure: Signature (r,s).

1: Randomly choose an integer k such that $1 \leq k \leq n-1$
2: Set the point $(x_1, y_1) = kP$ asserting it is an integer.
3: Set $r = x_1 \mod n$
4: **if** $r = 0$ **then**
5: Go back to step 1
6: **end if**
7: Let $s = k^{-1}(H(m) + d_A \cdot r) \mod n$
8: **if** $s = 0$ **then**
9: Go back to step 1
10: **end if**
11: Return signature (r,s)

Example 2. *Let's consider E (elliptic curve) over* $\mathbb{Z}_p$ *given by* $y^2 = x^3 + 7$*. The group of points on this curve are used to generate and verify ECDSA signatures for Bitcoin transaction processes [12].*

Algorithm 2 ECDSA: Verification

Require: Domain $D = (q, FR, a, b, P, n)$, Q_A (public key corresponding to the private key d_A), hash function $H(\cdot)$, message m to be signed, signature (r, s).

Ensure: Authentication or rejection.

if r or s is not in the range $[1, n-1]$ **then**
    Reject the signature
end if
Let $w = s^{-1} \mod n$
Let $u_1 = H(m) \cdot w \mod n$ and $u_2 = r \cdot w \mod n$
Set the point $X = (x_2, y_2) = u_1 P + u_2 Q_A$
if $X = \infty$ **then**
    Reject the signature
end if
Compute $v = x_2 \mod n$
if $v = r$ **then**
    Accept the signature as valid
else
    Reject the signature
end if

Remark 3. *ECC is widely used in modern cryptographic systems such as TLS, Bitcoin, and various secure communication protocols.*

Hashing

Hashing is a fundamental component of blockchain-based systems. It helps to secure the data and it ensures its integrity.

Definition 15. *An H function is called **hash** function if it takes a variable-length data block D as input and returns a fixed-length hash value given by $h = H(D)$ [13]*

Definition 16. *[13] A cryptographic hash function H is an algorithm that satisfies the following properties:*

- *The same input produces the same output.*
- *Infeasibility of finding x such that $H(x) = y$ for a given y.*
- *A small change in x can produce quite a distict y.*

- *It is infeasible to encounter x_1 and x_2, where $x_1 \neq x_2$, such that $H(x_1) = H(x_2)$.*

There are many families of cryptographic hash functions, but one of the most popular is SHA (Secure Hashing Algorithm). In fact, SHA-256 is the function used for hashing by Bitcoin and Ethereum.

Definition 17. *A **Merkle Tree** is an undirected, binary graph in which:*

- *Each leaf node represents a hash of a transaction or a data block.*
- *Each non-leaf node is the hash of its two child nodes.*

*The **Merkle Root**, which is the root (no parent nodes) of the Merkle Tree, represents with a single hash value the entire set of transactions of block.*

The Merkle Tree is an outstanding mechanism for efficient verification of the integrity of said transactions.

SHA256

Secure Hash Algorithm 256-bit, commonly known as SHA-256, is a cryptographic hash function, taking a message of arbitrary length and producing a 32-byte (256-bit) hash value.

Definition 18. *SHA-256 is a cryptographic hash function that processes a message M of arbitrary length less than 2^{64} bits and returns a 256-bit message digest [13].*

The algorithm operates on 512-bit message blocks and maintains a 256-bit state segmented into eight 32-bit words:

Algorithm 3 SHA-256 Algorithm [14]

Require: Message M of length l bits
Ensure: 256-bit hash value

Pad message to length multiple of 512 bits
Parse padded message into N 512-bit blocks: $M^{(1)}, M^{(2)}, ..., M^{(N)}$
Initialize hash values $H^{(0)}$ with first 32 bits of fractional parts of square roots of first 8 primes
for $i = 1$ to N **do**
    Initialize message schedule W_t:
    for $t = 0$ to 15 **do**
        $W_t \leftarrow M_t^{(i)}$ (32-bit block from message)
    end for
    for $t = 16$ to 63 **do**
        $\sigma_1 \leftarrow (W_{t-2} \ggg 17) \oplus (W_{t-2} \ggg 19) \oplus (W_{t-2} \gg 10)$
        $\sigma_0 \leftarrow (W_{t-15} \ggg 7) \oplus (W_{t-15} \ggg 18) \oplus (W_{t-15} \gg 3)$
        $W_t \leftarrow W_{t-16} + \sigma_0 + W_{t-7} + \sigma_1$
    end for
    Initialize working variables: $a \leftarrow H_0^{(i-1)}, b \leftarrow H_1^{(i-1)}, ..., h \leftarrow H_7^{(i-1)}$
    for $t = 0$ to 63 **do**
        $\Sigma_1 \leftarrow (e \ggg 6) \oplus (e \ggg 11) \oplus (e \ggg 25)$
        $Ch \leftarrow (e \wedge f) \oplus (\neg e \wedge g)$
        $T_1 \leftarrow h + \Sigma_1 + Ch + K_t + W_t$
        $\Sigma_0 \leftarrow (a \ggg 2) \oplus (a \ggg 13) \oplus (a \ggg 22)$
        $Maj \leftarrow (a \wedge b) \oplus (a \wedge c) \oplus (b \wedge c)$
        $T_2 \leftarrow \Sigma_0 + Maj$
        $h \leftarrow g$
        $g \leftarrow f$
        $f \leftarrow e$
        $e \leftarrow d + T_1$
        $d \leftarrow c$
        $c \leftarrow b$
        $b \leftarrow a$
        $a \leftarrow T_1 + T_2$
    end for
    Update hash values:
    $H_0^{(i)} \leftarrow a + H_0^{(i-1)}$
    $H_1^{(i)} \leftarrow b + H_1^{(i-1)}$
    $\vdots$
    $H_7^{(i)} \leftarrow h + H_7^{(i-1)}$
end for
return $H^{(N)} = (H_0^{(N)} \| H_1^{(N)} \| ... \| H_7^{(N)})$

$$H^{(i)} = (H_0^{(i)}, H_1^{(i)}, H_2^{(i)}, H_3^{(i)}, H_4^{(i)}, H_5^{(i)}, H_6^{(i)}, H_7^{(i)})$$

Where:

- $\ggg n$ denotes right rotation by n positions
- $\gg n$ denotes right shift by n positions

- $\oplus$ denotes bitwise XOR
- $\wedge$ denotes bitwise AND
- $\neg$ denotes bitwise NOT
- K_t are 64 constant words derived from cube roots of first 64 primes

The compression function will process each 512-bit block in 64 rounds, using the set of logical functions previously defined and modular additions to mix the message with the internal state. The hash value to be returned is the concatenation of the eight 32-bit words in the final state.

Blockchain

In essence, a blockchain can be understood as a sequence of blocks, where every block contains a collection of transactions.

Definition 19. *Let each block B_i be a tuple of the following structure:*

$$B_i = (H_{prev}, T, H_{merkle}, t, n)$$

where:

- *T is a list of transactions contained by the block.*
- *H_{merkle} is the Merkle tree's root hash assembled from the transactions in T, which provides integrity to the transactions.*
- *t is the timestamp, indicating when the block was created.*

- *n is the nonce, a value used to satisfy the proof-of-work condition.*
- *H_{prev} is the previous block's hash (links the blocks).*

It is important to highlight that there are components of the tuple that may be omitted and some that may be added (depends on the context of the blockchain).

Definition 20. *A **blockchain** is a sequence of the form:*

$$\mathscr{B} = (B_i)$$

where B_i represents the i-th block in the chain. Each block B_i is linked to its prior block B_{i-1} through a cryptographic hash.

Methodology

The proposed prototype blockchain-based voting system is a simple simulation on a CSV file, which contains voter information. This process is carried out on a single machine and port, and the votes occur sequentially. The idea behind this simulation is to understand the basic concepts of a blockchain-based voting system and to analyze the feasibility of solution it proposes.

Voter Data Structure

The simulation carried out uses a CSV file that contains voter information. Among the attributes per voter, we have:

- Voter ID: A unique identifier for each voter. In this case, it is the voter's ecuadorian identity card number.
- Name: Full name of the voter.
- Vote Preference: Binary choice between two candidates (A or B).
- Change Probability: It is a number between 0 and 1 that represents the probability of the voters changing their initial preference.
- Absence Probability: It is a number between 0 and 1 that represents the probability of the voters not taking part in the electoral process.

Transaction Creation and Verification

Each vote is represented as a transaction in the network. First, the CSV row for each voter is read. Depending on the voter's absence probability, the vote is created using the voter's

preference, and the change probability. Then, the vote is signed using the voter's private key (ECDSA), which will be generated using Algorithm 1 (the SECP256k1 curve, given by the equation $y^2 = x^3 + 7$ (the rest of the domain parameters can be found in [12]). The transaction is then generated, containing the signed vote.

Before a transaction (with the signed vote) is added to the pending pool, it has to be verified. The same curve that was used to generate the private key is used to get the public key. After, the voter's public key and the hashed transaction are used to validate the signature using Algorithm 2. Invalid transactions are rejected to maintain process integrity. Finally, the transaction is checked for duplicate votes using the voter's ID.

Block Structure

Each block $B = (H_{\text{prev}}, T, t, n, d)$ in the blockchain contains:

- H_{prev}: Hash of the previous block
- T: Single vote transaction
- t: Timestamp
- n: Nonce for PoW
- d: Difficulty target
- i: Block index in the chain

The blockchain $\mathscr{B} = (B_i)$ is defined as a sequence of these blocks. Since each block contains only one vote transaction, the transaction hash is basically its own merkle root.

Proof of Work (PoW)

The simulation implements a PoW mechanism where miners try to find a hash that satisfies certain conditions, or difficulty target. Once the transactions are verified, a candidate block is created. Nonetheless, this is not added to the blockchain until it surpases the difficulty target of the PoW. It is important to remember that the hashed message is the same every time you run it, but then how do we get a different output to try and solve the PoW algorithm? The answer is the nonce, a simple number or value that is added to the block's data, and changes completely the output of the hash (as discussed in Definition 16). Once a valid hash is found, the confirmed block is added to the blockchain.

As the difficulty of the PoW and number of participants increase, the need for computational power to solve the problem also increases. For this simulation, the difficulty target is defined as the number of leading zeros in the hash. This process ensures computational effort to add blocks, providing security against vote manipulation.

Overall Simulation Process

The simulation process can be summarized in these steps:

- Voter data from the CSV file
- Processes votes sequentially. A vote is created using the voter's preference and change probability.
- The transaction is generated, containing the signed voted.
- The verification process of ECDSA takes place, and after the transaction is added to the

pending pool.

- Proof of Work is performed, and if the hash satisfies the difficulty target, the block is added to the blockchain.
- Records voting results and blockchain statistics.

Security Considerations

The security of the proposed simulation relays on the following:

- ECDSA to sign and verify votes.
- PoW mechanism, preventing vote tampering.
- Unique voter identification, preventing double-voting.
- Immutable blockchain structure, ensuring vote integrity.

This approach helps to understand and visualize some basic components of blockchain technology, and the security features that it provides for voting processes.

Results and Discussion

For the purpose of the simulation and evaluation of the proposed system, a dataset of ten thousand voters was generated with the help of the Python programming language. The voter data was created using the `Faker` library for generating realistic voter names. Additionally, apart from the voter's preference, other aspects, such as change probability and absence probability, were introduced to mimic real life variability in the participants' behaviour.

Category	**Votes**	**Percentage**
Candidate A	4,513	45.13%
Candidate B	4,431	44.31%
Absence	1,056	10.56%
Total Eligible Voters	10,000	100%

Table 1: Election Results Summary

The results of the simulated election, as shown in Table 1, demonstrate a close race between the candidates, with Candidate A being the winner with 4,513 votes (45.13%) compared to Candidate B's 4,431 votes (44.31%). The voter turnout was relatively high at 89.44%, with only 1,056 voters (10.56%) not participating in the electoral process.

As anticipated, each vote was signed, verified, and successfully processed as a unique transaction in the blockchain. The correct implementation and functionality of the PoW mechanism ensures an e-voting system with high immutability. The ECDSA signing process for each vote transaction worked effectively in the conducted simulation, maintaining vote integrity and voter authentication. The blockchain structure maintained a verifiable and tamper-resistant chain of voting records, with each block containing a single vote transaction, timestamp, and previous block reference. Besides, the simulation demonstrated effectiveness to prevent double-voting, as the voter ID was validated for each vote transaction.

Overall, the results from the executed simulation indicate the first prototype of a simple, single-node e-voting system is works as expected. Moreover, it shows that the ideas behind blockchain open the possibility to enhance security measures, ensuring vote integrity and verification.

Conclusion

The use of blockchain-based technology in the e-voting systems represents the potential of providing solutions to the problems of security, transparency and integrity in the current electoral systems. This paper successfully and thoroughly discusses the mathematical background of the blockchain technology, with particular reference to elliptic curve cryptography and cryptographic hash functions, that guarantee the solidity of the voting system. This mathematical emphasis reflects the importance of deeply understanding the core concepts of this technology, to be able to correctly implement and manipulate the components of such a system. This research shows the usage of mathematical tools to transform the electoral processes, making democratic processes more secure, transparent, and accessible through technological advancement.

Future Work

While this mathematical understanding of blockchain is vital when the long-term goal is to contribute to our society with new and transparent methods for electoral processes, there are more areas where further research could lay the ground for important breakthroughs. Future work will focus on exploring alternatives for proof-of-work to upgrade the efficiency of the current system. A multi-node system will also be implemented, for scaling the scope of the system. Ultimately, our efforts will be directed towards creating a micro-system for e-voting (if possible for Universidad San Francisco de Quito student government elections). This micro-system will allow us to test its practical limitations, so that further refining in the system could be done, and even scale it up again to a broader context.

REFERENCES

[1] S. Haber and W. S. Stornetta, *How to time-stamp a digital document*. Springer, 1991.

[2] C. Dwork and M. Naor, "Pricing via processing or combatting junk mail," in *Annual international cryptology conference*. Springer, 1992, pp. 139–147.

[3] A. Back *et al.*, "Hashcash-a denial of service counter-measure," 2002.

[4] S. Nakamoto *et al.*, "Bitcoin," *A peer-to-peer electronic cash system*, vol. 21260, 2008.

[5] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, vol. 3, no. 37, pp. 2–1, 2014.

[6] S. King and S. Nadal, "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake," *self-published paper, August*, vol. 19, no. 1, 2012.

[7] T. W. Judson, *Abstract algebra: theory and applications*, 2020.

[8] L. C. Washington, *Elliptic curves: number theory and cryptography*. Chapman and Hall/CRC, 2008.

[9] M. Artin, *Algebra*. Pearson Education, 2011. [Online]. Available: https://books.google.com.ec/books?id=S6GSAgAAQBAJ

[10] D. Hankerson, A. Menezes, and S. Vanstone, "Guide to elliptic curve cryptography 2004 new york, ny."

[11] A. Demeri, W. Diehl, and A. Salman, *SADDLE: Secure Aerial Data Delivery with Lightweight Encryption*, 07 2020, pp. 204–223.

[12] M. Qu, “Sec 2: Recommended elliptic curve domain parameters,” *Certicom Res., Mississauga, ON, Canada, Tech. Rep. SEC2-Ver-0.6*, 1999.

[13] W. Stallings, *Cryptography and Network Security: Principles and Practice, Global Edition*. Pearson Education, 2022. [Online]. Available: https://books.google.com.ec/books?id=E7-GEAAAQBAJ

[14] K. K. Ting, S. C. Yuen, K.-H. Lee, and P. H. Leong, “An fpga based sha-256 processor,” in *Field-Programmable Logic and Applications: Reconfigurable Computing Is Going Mainstream: 12th International Conference, FPL 2002 Montpellier, France, September 2–4, 2002 Proceedings 12*. Springer, 2002, pp. 577–585.