Protocolo de votación electrónica basado en Blockchain y Visual Cryptography

Christofer Fabián Chávez Carazas

Arequipa, Perú

2018

Protocolo de votación electrónica basado en Blockchain y Visual Cryptography

Por
Christofer Fabián Chávez Carazas

Tesis presentada a la
Escuela Profesional de Ciencia de la Computación
de la
UNIVERSIDAD NACIONAL DE SAN AGUSTÍN
como requisito
para obtener el título profesional
de
Licenciado en Ciencia de la Computación

Universidad Nacional de San Agustín
Facultad de Ingeniería de Producción y Servicios
Escuela Profesional de Ciencia de la Computación

Protocolo de votación electrónica basado en Blockchain y Visual Cryptography

Tesis de graduación presentado por el bachiller Christofer Fabián Chávez Carazas en el cumplimiento de los requisitos para obtener el título profesional de Licenciado en Ciencia de la Computación.

Arequipa, 19 de Junio del 2018

Aprobado por:

Prof. Dr. Nombre Apellido Presidente	Prof. Dr. Nombre Apellido Secretario
Prof. Dr. Nombre Apellido Integrante	Prof. Dr. Nombre Apellido Externo Universidad del ABC

A Dios, por todo lo que me ha dado, a todos los profesores por sus enseñanzas y algunos amigos.

Índice general

Índice de figuras

Índice de cuadros

Agradecimientos

Resumen

El voto electrónico (*e-voting*) es un tema latente en nuestra actual sociedad; la tecnología avanza más rápido de lo que lo hace nuestro sistema democrático. Los protocolos y sistemas de voto electrónico deben adecuarse a las necesidades de la democracia, siendo las más relevantes la transparencia, la verificabilidad, la seguridad y la simplicidad. Al mismo tiempo, la tecnología *blockchain* está cambiando nuestra forma de ver la seguridad de los datos, gracias a características como la descentralización, la inmutabilidad, el anonimato y la auditabilidad. Además, existen infinidad de métodos y modelos criptográficos que ayudan a que la transferencia de datos sea más segura. *Visual Cryptography* es una alternativa efectiva para cifrar datos como imágenes y texto, y puede ser usada para dar una capa de seguridad más al sistema.

Utilizando estas tecnologías se puede cumplir con algunos requerimientos que el *e-voting* necesita, además de arreglar problemas, como el de las plataformas de votación maliciosas, presentes en otros protocolos. La propuesta de esta investigación es crear un nuevo protocolo de *e-voting* con la ayuda de un modelo de *blockchain* para la seguridad de los datos y el consenso, *Visual Cryptography* para mejorar la privacidad de los datos, y un protocolo criptográfico para el proceso de votación.

Abstract

Electronic voting (e-voting) is a latent issue in our current society; Technology advances faster than our democratic system does. The protocols and electronic voting systems must be adapted to the needs of democracy, the most relevant being transparency, verifiability, security and simplicity. At the same time, blockchain technology is changing the way we view data security, thanks to features such as decentralization, immutability, anonymity and auditability. In addition, there are many methods and cryptographic models that help make data transfer safer. Visual Cryptography is an effective alternative to encrypt data such as images and text, and can be used to give a layer of security to the system.

Using these technologies can meet some requirements that e-voting needs, in addition to fixing problems, such as the malicious voting platforms, present in other protocols. The proposal of this research is to create a new e-voting protocol with the help of a blockchain model for data security and consensus, Visual Cryptography to improve the privacy of data, and a cryptographic protocol for the process of vote.

Capítulo 1

Introducción

El voto electrónico (*e-voting)* es una solución tecnológica para realizar de manera rápida y efectiva los procesos electorales. En la actualidad existen diferentes sistemas [3] que implementan el *e-voting* de maneras distintas y con diferentes métodos que se tienen que amoldar adecuadamente a los requerimientos que, para tener una buena democracia, un proceso de votación convencional debería tener. La totalidad de estos requerimientos están especificados en [4], algunos de los más importantes son la transparencia, la seguridad, la igualdad y la verificabilidad; todas características de una buena democracia. Además, todo proceso electoral debe tener tres fases claramente diferenciadas: registro, autenticación y votación.

La criptografía es un campo de la ciencia basado en la matemática que nos ayuda a mantener seguro todo tipo de datos; texto, imágenes, música y videos. Cualquier método criptográfico transforma los datos, de tal manera que no puedan ser descifrados por gente malintencionada si es que los datos fueran interceptados. Con base en la criptografía, surgió la tecnología *blockchain*; una base de datos distribuida, inmutable, anónima y auditable [1]. Uno de los problemas que la tecnología *blockchain* resuelve de forma efectiva es el problema de los generales bizantinos [5]. Viéndolo en contexto, este problema plantea la dificultad de conseguir el consenso en sistemas distribuidos o, en otras palabras, evitar que nodos maliciosos o dañados influyan en el resultado del sistema. Actualmente hay diferentes algoritmos de consenso [6] que el blockchain puede

utilizar para resolver el problema, cada uno con sus pros y sus contras.

Otro modelo criptográfico muy utilizado para cifrar imágenes es el *Visual Cryptography* [7], el cual, el modelo más simple, consiste en generar más imágenes mediante una operación XOR, llamadas *shares*. Estas no muestran ningún tipo de información al tenerse por separado, pero al volverlas a juntar se tiene la imagen original. La idea principal de *Visual Cryptography* es transmitir los *shares* de forma separada o por medios distintos para que ninguna persona malintencionada intercepte todos los *shares*, o en el caso de que intercepte uno, lo descarte creyendo que es información basura.

Para el *e-voting*, se propuso un protocolo criptográfico para la fase de votación [8], que tiene el problema de las plataformas de votación maliciosas, en otras palabras, el protocolo no puede distinguir si una plataforma esta insertando votos válidos o fraudulentos a la base de datos. Este problema puede ser resuelto implementando un modelo *blockchain* junto con el protocolo criptográfico, además de añadir más seguridad a los datos y cumplir algunos requerimientos del *e-voting*. Pero aún así, en este tipo de sistemas el *blockchain* puede tener algunos problemas: Al ser una red distribuida que envía datos entre sus nodos, existe la posibilidad de interceptarlos [9]. Esto puede ser un problema con los datos sensibles generados en los procesos de registro y autenticación, pero puede ser resuelto fácilmente añadiendo una capa más de seguridad a la transmisión de estos datos hacia el *blockchain* con un modelo de *Visual Cryptography*.

La propuesta de esta investigación es crear un nuevo protocolo de *e-voting* con la ayuda de un modelo de *blockchain* y *Visual Cryptography*, para la seguridad y privacidad de los datos, el consenso y la detección de nodos fraudulentos, y un protocolo criptográfico descrito en [8] para el proceso de votación.

1.1. Contribuciones

El aporte de esta investigación resolver el problema de las plataformas de votación maliciosas y mejorar la transparencia del protocolo criptográfico de votación descrito en [8] con un modelo de *blockchain*, y aumentar la privacidad de los datos con *Visual Cryptography*.

1.2. Planteamiento del problema

En una sociedad que va creciendo día a día, cada vez se vuelve más complicado y costoso realizar procesos electorales ordinarios. Gracias a la tecnología, hoy en día podemos agilizar estos procesos con los sistemas de *e-voting*. El *e-voting* tiene que cumplir con los requerimientos que la democracia dicta [4]. De entre todos estos, los más importantes son la seguridad, la transparencia y la verificabilidad. Para lograr mantener la seguridad, el *e-voting* se basa en la criptografía y en la seguridad de redes. Una propiedad fundamental que debería tener un sistema de *e-voting* para que sea seguro es la llamada privacidad eterna. La transparencia es un requerimiento que entra en conflicto con la privacidad. Para que un sistema *e-voting* funcione tiene que haber un equilibrio entre lo que se va a mantener como privado y lo que se va a mostrar al público en general.

La verificabilidad es uno de los requerimientos más complicados de resolver; las soluciones acarrean más problemas. Algunos sistemas de *e-voting* [10] [11] logran la verificabilidad proporcionando un recibo a los votantes, para poder verificar los resultados finales. Esto puede ser contraproducente. Supongamos que una persona obliga a otra a votar por un candidato en particular, con el recibo se podría verificar por quién ha votado la segunda persona, y facilitaría el trabajo de la primera. Ante este problema surgió la necesidad de que los sistemas de *e-voting* sean resistentes a la coerción y libres de recibos [12].

El protocolo criptográfico de votación descrito en [8] cumple con la propiedad de la libertad de recibos y cumple parcialmente la propiedad de la resistencia a la coerción, pero no abarca la transparencia y tiene el problema de las plataformas de votación maliciosas. Este problema puede ser arreglado aplicando *blockchain*, la cual es una tecnología de base de datos distribuida que logra que los datos sean inmutables. En una *blockchain*, los nodos de la red son los encargados de verificar la integridad de los datos y de identificar a los posibles nodos maliciosos o dañados en la red, gracias a diferentes algoritmos de consenso. Otro problema encontrado en el protocolo es el control que pueden tener las organizaciones encargadas sobre los votos emitidos por los votantes. Los votos son cifrados con una única llave privada en posesión de las organizaciones

encargadas, dando posibilidad a que estos puedan descifrar y ver los votos en cualquier instante. Esto puede ser resuelto parcialmente usando *Visual Cryptography* generando más de una llave privada que puedan ser repartidas a los encargados principales y estos sólo podrán descifrar los votos si juntan las llaves.
Por lo tanto, la propuesta de esta investigación es proponer un nuevo protocolo de votación utilizando un modelo de *blockchain* con *Visual Cryptography* para dar una posible solución al problema de las plataformas de votación maliciosas y cumplir con mantener la seguridad, la transparencia y la verificabilidad del sistema.

1.3. Justificación

En nuestra actual sociedad, los procesos electorales tradicionales presentan problemas que afectan a los gobiernos, a las organizaciones encargadas y a los votantes. Cada vez se vuelve más costoso preparar el proceso y su realización se torna lenta. En las elecciones generales del 2011 se registró un incremento de la población electoral del 21 % con respecto a las elecciones generales del 2006 [13], y en las elecciones generales del 2016 se registró un incremento en la población electoral del 14.8 % respecto a las elecciones generales del año 2011 [14]. La población electoral tiende a crecer, y con el tiempo el proceso tradicional se volverá mucho más lento.

A las personas no les gusta gastar su tiempo yendo a su lugar de votación asignado para luego esperar en largas colas. Algunas personas no llegan a votar; en las elecciones generales del 2016 el 19 % de los electores hábiles no fueron a sufragar [15], causándoles complicaciones posteriores que no ocurrirían si todo el proceso fuera automatizado y de forma remota. Hoy en día existe la tecnología para crear sistemas de votación electrónica, pero no cumplen con los requerimientos y los estándares que un proceso de estas dimensiones debería tener. Además de cumplir con los votantes, manteniendo la seguridad y la facilidad del proceso, también se debe cumplir con las necesidades de la democracia [4], lo cual no es nada fácil de hacer y aún queda mucho camino por recorrer.

1.4. Objetivos

Objetivo general

- Desarrollar un nuevo protocolo de *e-voting* basado en un modelo *blockchain* y en *Visual Cryptography* con la ayuda de protocolos criptográficos para aumentar la seguridad del sistema.

Objetivos específicos

- Diseñar un modelo *blockchain* para el voto electrónico.
- Aplicar *Visual Cryptography* y protocolos criptográficos para aumentar la seguridad del sistema.
- Evaluar la seguridad del protocolo.

1.5. Estructura de la tesis

[...]

Capítulo 2

Estado del Arte

Existe un considerable número de investigaciones que se centran en resolver los diferentes problemas que presenta la votación electrónica. En [16] se muestra un nuevo sistema de votación electrónica: Du-Vote, que elimina la suposición de que los votantes deben confiar en computadoras de uso general, logrando así resolver el problema de las máquinas de votación maliciosas por parte de los usuarios. Esta propuesta consigue la privacidad y la seguridad de los votos gracias al hardware criptográfico. Por otro lado, ellos dan por hecho que el servidor que verifica y valida los votos se encuentra seguro, y no muestran una forma de verificar si el servidor está haciendo o no correctamente su trabajo, por mal funcionamiento o estar en posesión de una persona malintencionada.
En [17] proponen TRVote, un nuevo sistema de votación electrónica basado en máquinas de votación DRE para asegurar la confiabilidad. Mediante el DRE propuesto, su método evita el problema de las plataformas de votación maliciosas, pero su método, al igual que el anterior, se basa en hardware criptográfico, y producir los DRE puede llegar a ser demasiado costoso.
En [18] intentan verificar la integridad de los resultados electorales y mejorar la transparencia en el proceso de votación con una mayor participación de la gente. En su protocolo, Você Fiscal, lo logran emitiendo un recibo, para que luego los votantes le tomen una fotografía y lo envíen mediante una aplicación móvil a una base de datos de resultados independientes. Luego, verifican los resultados comparando la base de

datos de resultados independientes con la base de datos de resultados oficiales. Esta propuesta, al no ser libre de recibos, es susceptible a los ataques de coerción, y además no garantiza la seguridad del sistema ante posibles maquinas de votación maliciosas.
En [19] proponen un método estadístico para verificar la integridad del sistema luego del proceso de votación. El modelo proporciona evidencia convincente de que el protocolo criptográfico no ha sido atacado durante el proceso de votación. Se basa en la suposición de que la mayoría de los votantes están dispuestos y son capaces de usar el protocolo criptográfico según sea necesario. Este modelo puede ser utilizado en diferentes protocolos de votación electrónica para verificar si el sistema se encuentra íntegro, pero este análisis se realiza luego de acabado el proceso de votación, no puede dar una alerta durante el proceso.
En [20] se propone un protocolo criptográfico de votación que cumple con la verificabilidad y la seguridad del sistema, además cumple con ser resistente a la coerción y libre de recibos. Este protocolo permite que los votantes puedan votar más de una vez. Los votantes pueden eliminar su voto o cambiarlo, pero si llegan a votar por dos candidatos diferentes, el voto es anulado en el proceso de conteo de votos. El problema de esto es que el sistema se pueda inundar con demasiados votos y hace que la eficiencia del sistema baje o caiga completamente. Además, mantienen el problema de las máquinas de votación maliciosas.
En [21] se propone un modelo para comparar protocolos de votación para poder hacer un análisis y luego poder hacer una proposición de cual tipo de sistema de votación es mejor usar para un contexto en particular.
En el estado del arte también hay investigaciones que proponen usar un modelo de *blockchain* para la votación electrónica. En [22] utilizan un modelo de *blockchain* basado en el *blockchain* de BitCoin. En resumen, el protocolo verifica el voto, actualiza la base de datos, crea un bloque nuevo y realiza el broadcast del nuevo bloque. La diferencia con el modelo de *blockchain* tradicional es que este protocolo no utiliza un algoritmo de consenso, por lo tanto, no resuelve de manera eficiente el problema de las máquinas de votación maliciosas. En [23] proponen un sistema de votación electrónica utilizando los contratos inteligentes que ofrece el *blockchain* de Ethereum. Esta propuesta amolda su protocolo al *blockchain* de Ethereum que ya se encuentra implementado y que es

Sistema/Protocolo de votación	**Basado en**	**Ventajas**	**Desventajas**
Du-Vote [16]	Hardware criptográfico	Evita máquinas de votación maliciosas El hardware aumenta la seguridad y privacidad	No asegura los servidores
TRVote [17]	Máquinas DRE	Evita máquinas de votación maliciosas	El hardware puede ser costoso y difícil de usar
Você Fiscal [18]	Comparación por *crowdsourcing*	Aumenta la transparencia Mayor participación ciudadana	Ataques de coerción Máquinas de votación maliciosas
An experiment on the security of the Norwegian electronic voting protocol [19]	Modelo estadístico	Detecta ataques terminado el proceso de votación Se acomoda a diferentes protocolos de votación	No detecta ataques durante el proceso de votación
Blockchain Based E-Voting Recording System Design [22]	Blockchain de Bitcoin (público)	Seguridad en los datos Transparencia Persistencia	No resuelve el consenso Máquinas de votación maliciosas
A Smart Contract for Boardroom Voting with Maximum Voter Privacy [23]	Blockchain de Ethereum (público)	Seguridad en los datos Transparencia Resuelve el consenso	La votación depende del costo del Ether El costo del Ether es muy cambiante

Tabla 2.1: Comparación de los sistemas y protocolos de votación

usado por toda la Internet. El problema con esto es que cada contrato inteligente tiene un costo, que depende del costo actual del token de Ethereum, el Ether. Actualmente el Ether es una de las criptomonedas más importantes del mercado [24] y su precio está en constante cambio. Esto puede conllevar a un gran gasto si se implementa el protocolo anteriormente descrito. En la presente investigación, el modelo *blockchain* es el que se va a moldear al protocolo de votación y no va a tener ningún costo adicional que dependa del mercado de criptomonedas.

En la Tabla 2.1 se muestra un resumen comparativo de los sistemas y protocolos mensionados antes.

Capítulo 3

Marco Teórico

3.1. Voto Electrónico

Los sistemas de voto electrónico (*e-voting*) realizan el proceso convencional de votación mediante la tecnología. Estos sistemas abarcan varios temas: Interacción humano computador, criptografía, comunicación, redes computacionales y seguridad en redes y datos. Los sistemas de *e-voting* pueden ser de tres tipos: los sistemas remotos; donde todo el proceso se hace de forma remota por medio de una aplicación conectada a Internet en las computadoras personales de cada votante, los sistemas no remotos; que involucran todo tipo de interacción con el votante (máquinas de votación electrónica, sistemas de escaneo) que se hacen en un lugar de votación controlado, y la votación híbrida; que combina los dos tipos de *e-voting* anteriores.

3.1.1. Sistemas remotos

Los sistemas remotos generalmente son aplicaciones web que mandan el voto de los votantes a un servidor central previa identificación. Aquí juega un rol importante la seguridad de los datos cuando viajan al servidor central y la autenticación del votante, esto porque cualquier persona con acceso a internet puede entrar a la aplicación, y el objetivo es que no se puede suplantar a otras personas o que se pueda votar una segunda vez.

3.1.2. Sistemas no remotos

Los sistemas de este tipo más conocidos son los sistemas basados en máquinas de votación de Grabación directa electrónica (*Direct-recording electronic - DRE*). El diseño de estas máquinas es un tema de investigación en el área de Interacción humano computador [17]. La funcionalidad principal de un DRE es registrar el voto de los votantes, mandarlo a una localización central y, dependiendo del protocolo de votación, se imprime un comprobante.

3.1.3. Sistemas híbridos

Estos sistemas surgieron por la falta de Internet en algunos sectores. Este tipo de sistemas combina los sistemas remotos y los no remotos. El sistema es capaz de soportar tanto votos provenientes de las computadoras personales de los votantes como los votos provenientes de DREs. Dependiendo del sistema, los DREs que se implementan guardan los votos en un disco interno y luego son transferidos al servidor central, o se colocan computadoras personales con la aplicación ejecutándose localmente.

3.1.4. Requerimientos Constitucionales

Un sistema de *e-voting* tiene que implementar de manera eficiente el proceso de votación cumpliendo con los requerimientos de un estado democrático [4]. Estos requerimientos son: democracia, generalidad, libertad, igualdad, secretismo, franquedad.

Democracia

Es el más importante de los requerimientos y es el encargado de hacer que un sistema de *e-voting* tenga los mismos atributos que un proceso de votación convencional. Estos atributos son:

- **Transparencia:** El sistema de votación debe proporcionar todos los medios para convencer a los votantes de que se esté cometiendo fraude dentro o por el mismo sistema de *e-voting*.
- **Verificabilidad:** El sistema de votación debe proporcionan una manera de verificar los votos y los resultados finales del proceso de votación.

- **Rendición de cuentas:** El sistema de votación debe de registrar todas las operaciones relacionadas con el proceso de votación.
- **Confiabilidad:** El sistema de votación debe de asegurarse que el resultado final de las elecciones corresponda a los votos emitidos en el proceso de votación.
- **Seguridad:** Este punto abarca todo el sistema de votación. El sistema debe de ser seguro en todos los aspectos (base de datos, transmisión de datos, registro, autenticación, etc).
- **Simplicidad y accesibilidad:** El sistema de votación debe ser simple y fácil de usar para el votante.

Generalidad

Este requerimiento dicta que todos los votantes pueden participar en el proceso de votación y nadie debe ser excluido. Este es uno de los mayores problemas que enfrentan los sistemas de *e-voting* en general, ya que la tecnología no siempre está al alcance de todas las personas. Por esta razón, se dice que los sistemas de *e-voting* deben ser una alternativa al proceso convencional de votación.

Libertad

Este requerimiento se refiere a que el proceso de votación se debe llevar a cabo sin ningún tipo manipulación, coerción, o cualquier tipo de influencia. Se debe cumplir la libre expresión de las preferencias de los votantes, en el sistema de votación, esto es la posibilidad de emitir un voto nulo o en blanco. En el sistema de votación no debe aparecer propaganda de ningún tipo, mucho menos política. Otros requisitos que se deben cumplir son la resistencia a la coerción y la libertad de recibos, los cuales se explicará con mejor detalle en la Sección 3.1.5.

Igualdad

Para lograr este requerimiento se necesita cumplir con tres sub-requerimientos. Los candidatos deben ser considerados iguales en el sistema, esto quiere decir que son iguales

tanto en el *front-end* (las imágenes de los candidatos deben mostrarse de la misma forma) como en el *back-end* (cada voto vale como una para todos los candidatos). Los votantes deben ser considerados iguales, esto quiere decir que el voto de cada votante vale como uno. Esto también aplica en los sistemas híbridos, los votantes que utilicen en DRE deben ser iguales a los votantes que utilicen la aplicación por Internet. Y por último, los votantes sólo deben de ser capaces de dar un sólo voto y éste no puede ser modificado. La igualdad tiene mucho que ver con la transparencia de los datos, ya que el sistema debe ser capaz de convencer a las personas de que sus votos cuentan como uno para el candidato por el cual votaron.

Secretismo

Este requerimiento nos indica que los datos del votante (información personal y su voto) debe de mantenerse en secreto durante todo el proceso de votación, nadie, ni siquiera los organizadores ni personas del gobierno, deben de poder acceder a los datos del votante. Aquí también se habla de la necesidad de separar los procesos de registro y autenticación; éstos deben estar claramente definidos en el sistema. El secretismo entra en conflicto con la transparencia, ya que se torna difícil lograr la transparencia cuando no se tiene que mostrar los datos de los votantes.

Franquedad

Este requerimiento dicta que no debe haber ningún intermediario en el proceso de votación (Ninguna persona está autorizada para votar por otra persona). En el proceso de autenticación, se debe verificar que la persona no esté suplantando la identidad de otra. Otro sub-requerimiento que es necesario para que la franquedad se cumpla, es que los votos deben de ser guardados y contados correctamente.

3.1.5. Requerimientos de Seguridad

La presente tesis está centrada en aspectos de seguridad de los sistemas de *e-voting*. Los protocolos de seguridad y la criptografía mantiene la privacidad y la integridad de los datos de los votantes, y la integridad de todo el sistema. De esto depende que las

personas sigan teniendo confianza en este tipo de sistemas. Idealmente, una sistema de *e-voting* se quiere que sea privado, libre de recibos y resistente a la coerción.

Privacidad eterna

La mejor forma de mantener la privacidad es utilizando métodos criptográficos. En el proceso de votación, la forma más simple de mantener la privacidad es cifrando la información y el voto de cada uno de los votantes. Nadie, ni siquiera los organizadores ni los candidatos, debe poder descifrar los datos. Pero aún así existe una gran preocupación: un voto pueda ser vinculado a una persona y pueda ser descifrado décadas en el futuro con una tecnología avanzada. De este problema surgió la necesidad de que los sistemas de votación tengan privacidad eterna.

"Un esquema de votación tiene privacidad eterna si su privacidad no depende de suposiciones de dureza criptográfica" [25].

Resistente a la coerción

La coerción es cuando una persona, el coersor, incita, manipula o compra a un votante durante el proceso de votación con el fin de guiar el comportamiento del votante. En los procesos de votación, tanto tradicionales como electrónicos, existen diferentes formas de coerción: cuando se fuerza a la persona a votar por un candidato en especial (*forced vote*), cuando se fuerza a la persona a votar por un candidato aleatorio (*forced randomization*), cuando se fuerza a la persona a no votar (*forced abstention*), y cuando se fuerza a la persona a entregar sus credenciales (*forced surrender of credentials).* Por todo lo anterior, los sistemas de votación deben de ser resistentes a la coerción.

"Un esquema de votación es resistente a la coerción si cumple lo siguiente: Existe una estrategia para un votante forzado V tal que, para cualquier estrategia adoptada por el coersor C, V sea capaz de emitir su verdadero voto de manera que sea indistinguible de C, siguiendo, de igual manera, las instrucciones de C" [25].

Libre de recibos

Al querer resolver problemas de verificabilidad, algunos sistemas de votación electrónica [3] [18] [10] [11] utilizan recibos para verificar los resultados finales, pero esto puede

generar otros problemas. Si el protocolo de votación utiliza un recibo que no está muy bien diseñado, un coersor puede utilizar el recibo para verificar el voto del votante forzado.

"Un sistema de votación es libre de recibos si el votante es incapaz de probar cómo y por quién voto, aun así esté confabulado con el coersor y se desvía del protocolo para tratar de presentar una prueba" [25].

3.2. Preliminares criptográficos

3.2.1. Criptografía asimétrica

La criptografía asimétrica o la criptografía de clave pública [26] utiliza dos claves para enviar un mensaje. Una clave es privada, y es mantenida en secreto, y la otra clave es pública y puede ser mostrada a cualquier usuario. La clave pública no es suficiente para obtener, mediante ataques, la clave privada de un usuario. Existen dos tipos de criptografía asimétrica que difieren en la utilización de las claves:

- **Cifrado de mensaje:** El emisor cifra el mensaje con la clave pública del receptor, y el receptor descifra el mensaje con su clave privada.
- **Firma digital:** El emisor cifra el mensaje con su clave privada, y el receptor descifra el mensaje con la clave pública del emisor.

Existen diferentes métodos que utilizan criptografía asimétrica, pero en esta investigación sólo utilizaremos dos: RSA [27]; donde la seguridad radica en el problema de la factorización de números enteros, y ElGamal [28]; donde la seguridad radica en el problema del logaritmo discreto.

3.2.2. Cifrado Homomórfico

Un esquema de cifrado es homomórfico [29] cuando sobre una operación '$\star$', soporta la siguiente fórmula:

$$E(m1) \star E(m2) = E(m1 \star m2),\ \forall m1, m2 \in M,$$

done E es el algoritmo de cifrado y M es el conjunto de posibles mensajes.
Cuando un esquema de cifrado soporta o la suma o la multiplicación, se dice que es parcialmente homomórfico. Cuando un esquema criptográfico soporta tanto la suma como la multiplicación, se dice que es totalmente homomórfico.

3.2.3. Mezclado Criptográfico

El objetivo principal de un mezclador criptográfico es desvincular los valores de salida de los valores de entrada [8]. Dado un conjunto de entrada $Z = (z_1, ..., z_n)$, el mezclador aplica una función unidireccional con llave $f_{k_i} : Z \to Z$ a cada valor de entrada $z_i \in Z$, con una llave k_i perteneciente a un conjunto de llaves $K = (k_n, ...k_n)$. Luego, se permutan los resultados escogiendo una permutación aleatoria $\phi : \{1, ..., n\} \to \{1, ..., n\}$. La salida del mezclador es una lista $Z' = (z'_1, ..., z'_n)$ de valores $z'_j = f_{k_i}(z_i)$ con índices $j = \phi(i)$. Todo lo anterior se puede escribir con la siguietne fórmula:

$$Z' = shuffle^{\phi}_{f_K}(Z)$$

3.2.4. Grupo Cíclico Multiplicativo

Un grupo cíclico G_p de orden p, es un conjunto de $p - 1$ valores que pueden ser obtenidos por un sólo elemento g al cual se le llama generador. Un grupo cíclico multiplicativo es un grupo cíclico donde cada valor G_i se genera con la siguiente fórmula:

$$G_i = g^a \, mod \, p$$

para algún $a \in \{0, 1, ..., p - 1\}$.
En criptografía se utilizan grupos cíclicos de orden primo, p es un número primo.
Un grupo cíclico $H_q \subset \mathbb{Z}^*_p$ es sub-grupo de G_p cuando existe un factor común k de p y q.
Un grupo cíclico $H_q \subset \mathbb{Z}^*_p$ es sub-grupo de G_p de orden primo p cuando existe un factor común $k = (p - 1)/q$.

3.2.5. Esquema de Compromiso

Un esquema de compromiso [30] permite comprometerse a un mensaje oculto para que pueda ser revelado luego El esquema de compromiso es utilizado para verificar que el mensaje ha llegado intacto y evitar que ninguna de las dos partes, ni un tercera, haga modificaciones una vez el compromiso se haya efectuado. Uno de los esquemas de compromiso más utilizado es el esquema de Pedersen [31].
Dado un grupo cíclico $H_q \subset \mathbb{Z}_p^*$ subgrupo de un grupo cíclico G_p con orden primo p. Se obtienen los valores g y h de H_q tal que nadie conozca el valor de $log_g h$; estos valores son públicos. Se quiere comprometer el mensaje $s \in \mathbb{Z}_q$. Se escoje un valor $t \in \mathbb{Z}_q$ de manera aleatoria y se utiliza la siguiente fórmula:

$$E(s,t) = g^s h^t$$

Luego, el compromiso puede ser abierto revelando los valores de s y t. Se puede observar que si el mensaje s cambia, entonces el valor de $E(s,t)$ va a ser diferente al que se ha obtenido anteriormente, y, además, los valores $E(s,t)$, g y h no son suficientes para obtener el mensaje s.

3.2.6. Prueba de conocimiento cero

Una prueba de conocimiento cero [32] es un protocolo en donde un agente agente probador da todas las evidencias para probar a una agente verificador que una declaración es cierta sin revelar nada más que la verificación de la declaración. Un ejemplo simple es que un agente que tiene una clave pueda probar a otro agente que él tiene esa clave pero sin revelarla. Una prueba de conocimiento cero tiene las siguientes propiedades:

- **Totalidad:** La posibilidad de fallar en el protocolo que hace que un probador convenza a un verificador, debe ser lo más mínima posible.
- **Solvencia:** Si la declaración es falsa, el probador no debe poder convencer al verificador que la declaración sea verdadera.

- **Conocimiento cero:** El verificador sólo debe tener conocimiento de que la declaración es verdadera o falsa, no se le brinda más información.

Existen dos tipos de pruebas de conocimiento cero:

- **Pruebas de conocimiento cero interactivas:** Tanto el probador como el verificador deben estar presentes durante la ejecución del protocolo.
- **Pruebas de conocimiento cero no interactivas:** No es necesario que el probador y el verificado estén presentes durante la ejecución del protocolo. El probador genera las pruebas necesarias, para que luego el verificador las vea en otro momento.

3.3. Blockchain

Iniciando con una breve historia del *blockchain*, fue propuesto por una persona bajo el seudónimo de "Satoshi Nakamoto" en el artículo [33]. Inicialmente fue propuesto para ser usado en un sistema de pagos *peer-to-peer* cuyas transacciones podrían ser hechas por Internet sin la necesidad de que una entidad financiera que respaldara dichas transacciones; los mismos nodos del sistema son los encargados de validar cada transacción que se haga en el sistema.

Blockchain, en la práctica, es una tecnología de base de datos distribuida, en donde cada nodo es considerado de igual peso en la red. La base de datos se replica en cada uno de los nodos, y es trabajo de ellos mantener la integridad de los datos. *Blockchain* utiliza métodos criptográficos para mantener la seguridad y la privacidad en la red. Además, *blockchain* utiliza algoritmos de consenso para evitar que nodos dañados o mal intencionados puedan modificar o falsificar los datos.

El primer sistema que se implementó fue el BitCoin [33] creando una nueva e innovadora economía: Las criptomonedas. Hoy existen numerosos sistemas financieros con base en el *blockchain*: Ethereum [34], Ripple [35], Monero [36], entre otros. Actualmente, el *blockchain* es utilizado para varias aplicaciones muy distintas al mundo financiero [24], como en el sector salud [37], internet de las cosas [38], sistemas de reputación [39] y muchas otras más [24][1].

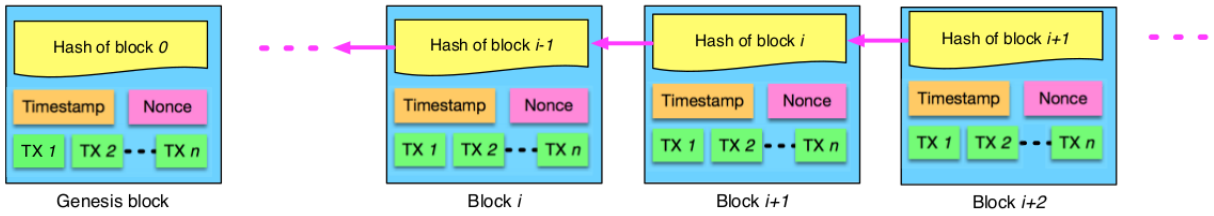

Figura 3.1: Ejemplo de la arquitectura de Blockchain [1]

3.3.1. Arquitectura del Blockchain

Blockchain es una cadena o secuencia de bloques, los cuales pueden guardar cualquier tipo de información dentro. En la Figura 3.1 se muestra un ejemplo de arquitectura del blockchain. En cada bloque se guarda un número n de datos o transacciones TX. Para poder identificar cada bloque, se juntan todos los datos del bloque y se les aplica una función hash criptográfica. El hash resultante tiene tres funcionalidades, funciona como un identificador del bloque, sirve para la verificación de los datos del mismo bloque, y sirve para la verificación de los datos del siguiente bloque que se agregue a la cadena.

Dado el conjunto de datos TX^i del bloque i, el hash h^i siempre va a ser el mismo si se aplica la misma función hash criptográfica a TX^i. Si se hace algún pequeño cambio a cualquier dato en TX^i, entonces h^i cambiaría completamente. De esta forma es bastante fácil saber si los datos de un bloque fueron modificados, sólo se tiene que verificar la integridad del hash. Los bloques, además de tener su propio hash, tienen el hash del bloque anterior de la cadena, también llamado bloque padre del bloque actual. Dado el conjunto de datos TX^{i+1}, el hash h^{i+1} que va a identificar al bloque $i+1$, va a ser el resultado de aplicar la función hash criptográfica a la unión de h^i y TX^{i+1}. Esto quiere decir que h^{i+1} depende también de h^i. En otras palabras, si algún bloque anterior a $i+1$ es modificado, entonces el hash h^{i+1} cambiaría completamente. Por consiguiente, sólo verificando el hash del último bloque de la cadena, se puede verificar la integridad de todos los datos del *blockchain*.

El *Timestamp* contiene la marca de tiempo del momento en que se creó el bloque, y el

Nonce es un valor utilizado en los algoritmos de consenso, los cuales se explicarán en la Sección 3.3.4.

3.3.2. Red Blockchain

La red de *blockchain* es *peer-to-peer* y distribuida. Cada nodo tiene el mismo peso en la red y está virtualmente conectado con todos los otros nodos de la red. La base de datos es replicada en cada nodo. Dependiendo del uso que se le vaya a dar al *blockchain*, los nodos deben guardar toda la cadena o partes de ella. Por ejemplo, en el caso del BitCoin, los nodos deben guardar toda la cadena, ya que deben verificar que el monto de las nuevas transacciones concuerde con las anteriores. Pero en otros casos, como por ejemplo datos en la nube, gracias a la arquitectura del *blockchain*, no sería necesario que los nodos guarden todos los bloques, sólo sería necesario guardar el último bloque de la cadena, ya que lo único que se verificaría sería la integridad de los datos. Cualquier nodo puede crear un bloque, pero éste es añadido a la cadena cuando es validado por todos los demás nodos. Esta validación se logra de manera efectiva gracias a los algoritmos de consenso.

3.3.3. Privacidad y firma digital

Una característica del *blockchain* es que mantiene la privacidad tanto de los nodos que verifican la cadena como de los nodos que participan en ella. Para lograr esto, *blockchain* utiliza la criptografía asimétrica. En resumen, la criptografía asimétrica o de llave pública, consiste en generar dos llaves: una pública y otra privada. El mensaje es cifrado con la llave privada y es descifrado con la llave pública. En el *blockchain*, para cada nodo en la red se crean las dos llaves, y los nodos son identificados por la llave pública.

Para poder verificar que una transacción en la cadena fue escrita por el un usuario se utiliza una firma digital. Un usuario a hace una transacción al usuario b. El usuario a obtiene el hash asociado a la transacción que acaba de hacer, lo cifra con su llave privada y envía la transacción con el hash cifrado al usuario b. El usuario b descifra el hash con la llave pública del usuario a verifica si el hash concuerda con la data enviada. De esta forma se logra verificar la identidad de los nodos que hacen transacciones.

3.3.4. Algoritmos de Consenso

El *blockchain*, como toda red distribuida que realiza tareas, tiene que resolver el problema de los generales bizantinos [5]. En la investigación presentan un problema de los sistemas distribuidos mediante una analogía. Un grupo de generales bizantinos quiere conquistar una ciudad enemiga. Su plan es posicionarse en diferentes puntos de la ciudad para analizar si es más factible atacar o retirarse, cada uno de los generales da su opinión y se realiza un consenso para poder tomar una decisión. El problema está cuando existen generales traidores y más cuando hay más generales traidores que leales. Los algoritmos de consenso son utilizados para resolver el consenso en redes distribuidas y para evitar que nodos maliciosos o dañados se apoderen de la red. En el modelo original de *blockchain* se propuso un algoritmo de consenso que resolvió estos problemas: *Proof of Work* [33].

El algoritmo de *Proof of Work* nos pone la regla de que los hash de todos los bloques comiencen por un número determinado de ceros. Para esto sirve el campo *Nonce* visto en la Sección 3.3.1. Este valor se agrega al final de los datos y se lo calcula de tal forma de que el hash resultante tenga al inicio los ceros determinados en ese momento. Esto hace que, para poder crear un bloque, necesites calcular el *Nonce*. Dependiendo del número de ceros, la dificultad de crear un bloque aumenta o disminuye. Por lo tanto, para poder agregar nodos fraudulentos a la red, se necesita tener un poder computacional mayor que la mitad de toda la red.

Dependiendo del contexto en el que se esté usando el *blockchain* existen varios algoritmos de consenso [6].

3.4. Visual Cryptography

Visual Cryptography fue introducido por Naor y Shamir [40]. Ellos propusieron un esquema criptográfico basado en la visión humana. El modelo está pensado para cifrar imágenes binarias, pero puede ser adaptado para cualquier tipo de información (texto, imágenes, videos). La idea principal del modelo es generar un número determinado de imágenes, llamadas en la literatura *shares*, a partir de la imagen original. Los *shares* por separado son, a simple vista, información basura, pero cuando se juntan y se

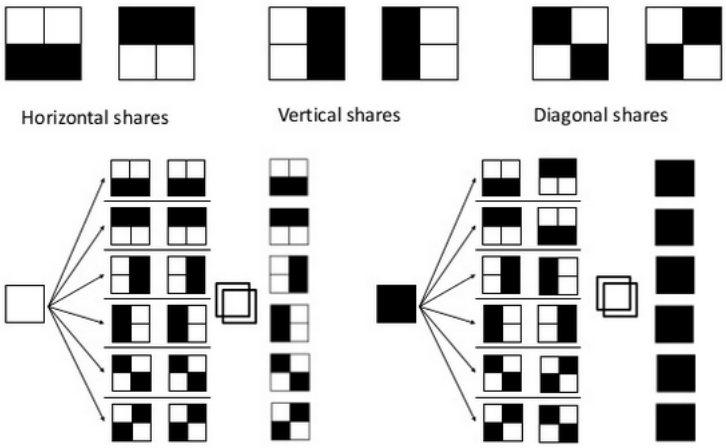

Figura 3.2: Combinación de píxeles del *Visual Cryptography* [2]

sobreponen, se obtiene la imagen original. Existen dos tipos de modelos:

- **Modelo (n,n):** El sistema genera n *shares* ($n \geq 2$) y todos los *shares* son necesarios para reconstruir la información original.
- **Modelo (k,n):** El sistema genera n *shares* ($n \geq 2$) y sólo son necesarios k *shares* ($2 \leq k \leq n$) para reconstruir la información original.

En imágenes binarias, los píxeles negros son los que contienen la información. En la Figura 3.2 se muestran las divisiones de los píxeles de la imagen original, de los dos *shares* generados, y de la imagen original después del descifrado. En este caso, al descifrar, sólo los píxeles negros son reconstruidos completamente, mientras que los píxeles blancos son reconstruidos y descartados con basura.

El método en [40] fue propuesto originalmente para cifrar imágenes y que la misma persona lo descifre, como si tuviera los dos *shares* impresos y los superpusiera a contra luz para descifrar la imagen original. Cuando no se necesita de la intervención humana para el descfrado, entonces no es necesario hacer las divisiones de píxeles.
Para un modelo (2,2), uno de los *shares* es generado de forma aleatoria. El segundo es generado aplicando una operación XOR entre la información original y el *share* creado. Se envían los *shares* por el medio de forma separada, para evitar que sean interceptados juntos. El destino recibe los dos *shares* y recupera la información original haciendo una operación XOR entre los dos *shares*. Este esquema criptográfico no necesita de

operaciones matemáticas complejas, así que no se necesita de hardware especializado o de última generación para poder utilizar el esquema.

Capítulo 4

Propuesta

En esta parte de la tesis se va a detallar el protocolo de votación electrónica propuesto. En la Figura 4.1 se muestra el proceso del protocolo de votación electrónica propuesto.

La primera parte es la preparación del proceso de votación. Aquí se generan las credenciales públicas y privadas para cada votante habilitado para votar. Se le hace entrega de la credencial privada al votante que le servirá posteriormente para realizar su voto. En la fase de votación el votante hace uso de su credencial privada y pública para realizar su voto. Con el algoritmo de cifrado de ElGamal se cifran las credenciales y el voto realizado. El voto cifrado es dividido en n *shares* mediante *Visual Cryptography*. Cada *share* es guardado dentro de un blockchain con las credenciales cifradas del votante. En el sistema hay un blockchain público donde se va a guardar un *share*; que va a servir para que el votante verifique si su voto fue registrado, y $n-1$ blockchains privados con el resto de los *shares*. Al finalizar el proceso de votación, se obtienen los *shares* de cada votante y se regenera el voto con Visual Cryptography. Luego, se realiza el conteo de los votos utilizando prueba de conocimiento cero; para validar los votos, y un mezclado criptográfico para realizar un mejor conteo sin escrúpulos. El resultado final es la sumatoria de votos para cada candidato.

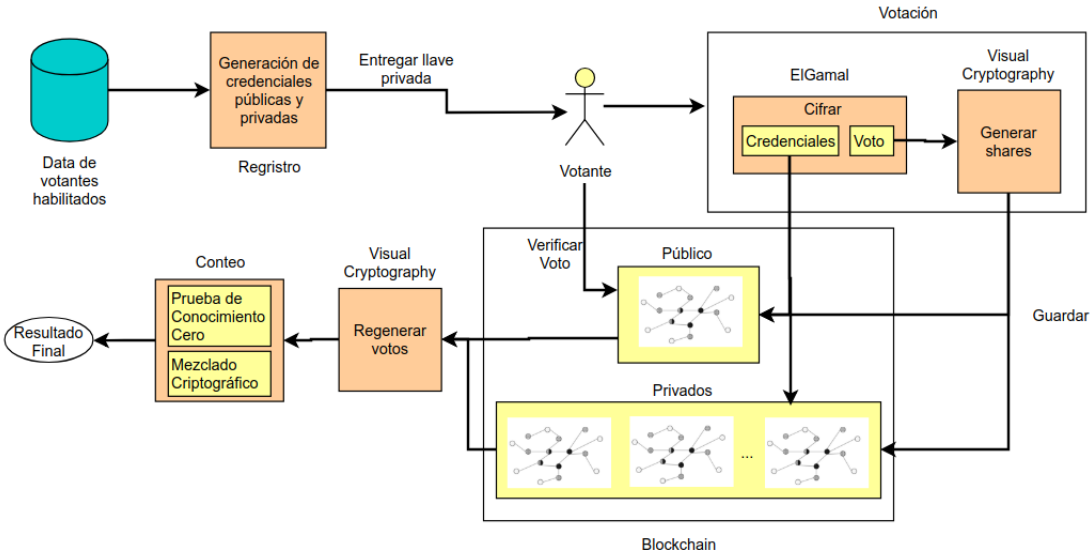

Figura 4.1: Proceso del protocolo de votación electrónica propuesto

4.1. Preparación

Para comenzar con la preparación del proceso de votación, se crea un grupo cíclico multiplicativo $G_q \subset \mathbb{Z}_p^*$ de orden primo q, subgrupo del grupo de enteros de orden primo p. Se obtienen dos generadores $g_1, g_2 \in G_q$, y se obtienen valores aleatorios $h_1, h_2 \in G_q$. El siguiente paso es obtener la información concerniente a los votantes habilitados para votar desde la base de datos de la organización encargada del proceso de votación. El conjunto $V = \{V_1, ..., V_m\}$ está conformado por los m votantes habilitados. Para cada votante V_i se obtiene la credencial privada $(\alpha, \beta, \gamma, \delta)_i \in G_q \times G_q \times G_q \times G_q$ de forma aleatoria, y se genera la credencial pública $u_i = h_1^{\gamma} h_2^{\delta}$.

Las credenciales públicas son vinculadas a cada votante, obteniendo una tupla (V_i, u_i). El conjunto de estas tuplas $PE = \{(V_1, u_1), ..., (V_m, u_m)\}$ contiene las credenciales públicas vinculadas a cada votante habilitado para votar, en otras palabras el conjunto PE actúa como el padrón electoral del proceso de votación, y es de dominio público. Los votantes pueden obtener su llave pública y verificar su participación en el proceso de votación revisando el conjunto PE. Las credenciales privadas son entregadas a cada votante por una canal seguro y borradas de cualquier lugar en donde hayan sido guardadas.

Luego, se obtiene la información de los candidatos para formar un conjunto $C =$

$\{C_1, ..., C_{n+1}\}$ con los n candidatos que participarán en las elecciones y un valor C_{n+1} para los votos nulos. Para cada candidato C_i, se genera un valor aleatorio $c_i \in G_q$, que va a actuar como identificador del candidato C_i. Los identificadores de los candidatos van a formar el conjunto $IC = \{c_1, ..., c_{n+1}\}$. Por cuestiones de verificabilidad, se vincula el identificador de candidato con la información de su respectivo candidato para formar una tupla (C_i, c_i). El conjunto de tuplas $PC = \{(C_1, c_1), ..., (C_{n+1}, c_{n+1})\}$ se hace de domino público. El conjunto PC sirve para que el público en general pueda verificar los identificadores de candidatos que el sistema de votación electrónica va a usar, en pocas palabras, este conjunto es para fines de verificabilidad, no es usado en el sistema. Por último se obtienen dos generadores $e_1, e_2 \in G_q$ que van a actuar como generadores electorales y también son de dominio público junto al conjunto PE y PC. Para el proceso de descomposición del voto mediante *Visual Cryptography* se genera de manera aleatoria un conjunto de *shares* $SO = \{so_1, ..., so_{n_o}\}$ donde n_o es el número de los organizadores encargados, y se le hace entrega de un *share* a cada organizador encargado. Estos *shares* son utilizados para descomponer los votos para luego guardarlos en el blockchain. Su principal objetivo es de asegurar de que ningún organizador por si solo pueda reconstruir los votos. En el proceso de conteo, para poder reconstruir los votos, se necesita de la presencia de todos los organizadores con sus respectivos *shares*.

4.2. Proceso de votación

Luego de obtener y validar todos los valores necesarios, se procede a iniciar el proceso de votación. Este proceso consta de dos partes: el cifrado del voto mediante ElGamal y la descomosición del voto cifrado mediante *Visual Cryptography*.

4.2.1. Cifrado mediante ElGamal

Antes de entrar en detalle al proceso de votación, primero se va a definir dos instancias distintas del cifrado de ElGamal, una estándar y otra exponencial. Se obtiene un generador $l \in G_q$ que va a ser usado en las dos instancias. Igualmente, se obtiene de manera aleatoria el valor de la llave privada $x \in G_q$, y se calcula el valor de la llave pública $y = l^x$. El valor de estas dos llaves es usado en las dos instancias y en todo el

proceso de votación, y la llave pública y es de dominio público. La instancia de ElGamal estándar se va a escribir de la forma $E = enc_y^{\times}(m, r) = (l^r, my^r)$ donde $m \in G_q$ es el mensaje a cifrar, $r \in G_q$ es un valor aleatorio y E es el mensaje cifrado representado por la tupla $(a, b) = (l^r, my^r)$. Para el descifrado se utiliza $m = dec_x^{\times}(E) = ba^{-x}$.
La instancia de ElGamal exponencial se va a escribir de la forma $E = enc_y^{+}(m, r) = (l^r, l^m y^r)$ donde $m \in G_q$ es el mensaje a cifrar, $r \in G_q$ es un valor aleatorio y E es el mensaje cifrado representado por la tupla $(a, b) = (l^r, l^m y^r)$. Para el descifrado se utiliza $m = dec_x^{+}(E) = log_l(ba^{-x})$.
El votante V_i ingresa al sistema de votación con su credencial privada. Se genera una credencial electoral $\hat{u_i} = e^{\alpha}e^{\beta}$ para el votante V_i y es enviada para verificar si es que ya hay un voto registrado, pendiente de validación o no hay ningún voto enlazado a esa credencial. Luego, realiza su voto $v_i \in IC$ de acuerdo al candidato de su preferencia. La credencial electoral $\hat{u_i}$ es cifrada con la instancia de ElGamal estándar y se obtiene $E_{u_i} = enc_y^{\times}(\hat{u_i}, r_i^u)$ con un valor aleatorio $r_i^u \in G_q$. El valor de r_i^u es la credencial de verificación que va a ser entregada al votante una vez que el voto ha sido enviado, verificado y guardado en el blockchain. El voto v_i se cifra con la instancia de ElGamal exponencial y se obtiene $E_{v_i} = enc_y^{+}(v_i, r_i^v)$ con un valor aleatorio $r_i^v \in G_q$.

4.2.2. Descomposición mediante Visual Cryptography

Para el protocolo propuesto usa un modelo (n, n) de *Visual Cryptography*. El valor de n denota el número de *shares* que se van a generar a partir del mensaje, y va a ser necesario de los n *shares* para poder reconstruir el mensaje original. En el protocolo propuesto, $n = n_o + b_n$ donde n_o es el número de *shares* de los organizadores encargados, y b_n es el número de cadenas de bloque que el sistema va a manejar (la estructura y funcionamiento del blockchain van a ser explicados en la siguiente sección). El modelo de *Visual Cryptography* recibe como entrada el voto cifrado E_{v_i} y el conjunto de *shares SO* y genera otro conjunto de *shares* $S_i = \{s_1^{v_i}, ..., s_{b_n}^{v_i}\}$ con b_n elementos. Cada *share* $s_j^{v_i}$ es vinculado con la credencial electoral del votante cifrada E_{u_i} formando así una tupla $(E_{u_i}, s_j^{v_i})$. El conjunto de estas tuplas $S_i' = \{(E_{u_i}, s_1^{v_i}), ..., (E_{u_i}, s_{b_n}^{v_i})\}$ es transmitido por internet hacia el blockchain.
Aquí se puede observar que los datos S' transmitidos desde la computadora del votante

hacia el blockchain se encuentran totalmente cifrados. Si el atacante logra interceptar una de las tuplas, no va a poder descifrar las credenciales sin la llave privada, y el *share* contenido en esa tupla es inservible sin los otros *shares*. Aun así, si el atacante logra interceptar todo los datos de S' y reconstruye el voto cifrado E_{v_i}, no va a poder descifrarlo sin la llave privada.

4.3. Blockchain

En esta sección se va a presentar el modelo de blockchain propuesto.

4.3.1. Arquitectura del blockchain propuesto

Como un blockchain simple, el blockchain propuesto es una cadena bloques, donde cada bloque es identificado por un hash y hacer referencia a un bloque anterior (Ver Sección 3.3). Además, el bloque contiene una lista T conformada por tuplas $(E_{u_i}, s_j^{v_i})$ que contienen la credencial electoral y un *share* del voto de un votante V_i. El hash que identifica al bloque es generado a partir de la unión de la lista T y el hash del bloque anterior. Se agrega un *Timestamp* que contiene la marca de tiempo del momento en que se creó el bloque y, a diferencia de los bloques de un blockchain simple, no contiene el valor de *Nonce* ya que se utiliza un algoritmo de consenso distinto al *Proof of Work* de un blockchain convencional.

4.3.2. Topología de la red blockchain

En la red hay un blockchain público b_p y $b_n - 1$ blockchain privados que van a formar el conjunto de blockchains $B = \{b_p, b_{q_1}, ..., b_{q_{b_n-1}}\}$. Una tupla aleatoria del conjunto S_i es guardada en b_p, y las restantes se guardan cada una en un blockchain privado b_{q_i} distinto. El b_p es de dominio público, y va a servir para que un votante pueda verificar si su voto ha sido registrado correctamente. Para lograr esta verificación, el votante vuelve a generar su credencial electoral $\hat{u_i} = e^{\alpha}e^{\beta}$ con su credencial privada. Luego, cifra su credencial con la instancia de ElGamal estándar y obtiene $E_{u_i} = enc_y^{\times}(\hat{u_i}, r_i^u)$ donde el valor de r_i^u es la credencial de verificación que se le dio al votante al momento de realizar el proceso de votación, y verifica si E_{u_i} se encuentra en b_p.

Nótece que el valor de la credencial de verificación r_i^u puede ser considerado considerado como un recibo que se emite luego de que un votante realiza su voto. La definición de "Libertad de recibos" (Ver Sección 3.1.5) nos dice que el votante debe de ser incapaz de probar cómo y por quién voto a un coersor. Con la llave privada y con la credencial de verificación, el votante es capaz de probar que votó o no, pero es incapaz de probar por quién votó ni saber tampoco si es voto nulo o no, esto porque en el blockchain público, la credencial electoral cifrada del votante esta enlazada a un *share* del voto cifrado del votante, el cual es insuficiente para revelar todo el voto, sin contar además de que este no va a poder ser descifrado sin la llave pública.
Una de las razones de el por qué separar el voto cifrado de un votante en un número determinado de *shares* es para mantener la privacidad eterna (Ver Sección 3.1.5). Para la privacidad eterna se presupone que se ha encontrado una forma de descifrar de forma eficiente cualquier mensaje cifrado con una instancia de ElGamal. Si el voto hubiera sido guardado en el blockchain sin ninguna subdivisión, entonces el votante podría fácilmente descifrarlo y probar al coersor por quién ha votado, pero con la división del voto cifrado en *shares*, esto no sería posible, ya que necesitaría de los otros *shares* de los blockchain privados y de los *shares* de las organizaciones encargadas para poder rearmar el voto cifrado y recién poder descifrarlo. La seguridad de los blockchain privados recae en los servidores en donde éstos van a ser guardados y en cómo las organizaciones encargadas mantengan seguros sus *shares*.

La topología de la red esta basada en la red de [41]. La topología de red del blockchain propuesto se muestra en la Figura 4.2. En la topología existen tres tipos de nodos que se pueden atachar a la red: los nodos comisionados, los nodos de sufragio y los nodos votantes.

Nodos comisionados

El conjunto de nnc nodos comisionados $NC = \{nc_1, ..., nc_{nnc}\}$ contiene los nodos más importantes de la red, ya que son los encargados de organizar el consenso para validar los votos y los bloques que los nodos de sufragio generen. También son los encargados de guardar la totalidad de los blockchains. Cada nodo nc_i va a guardar un blockchain $b \in B$; esto se define por la función $block(nc_i) = b : NC \rightarrow B$. El

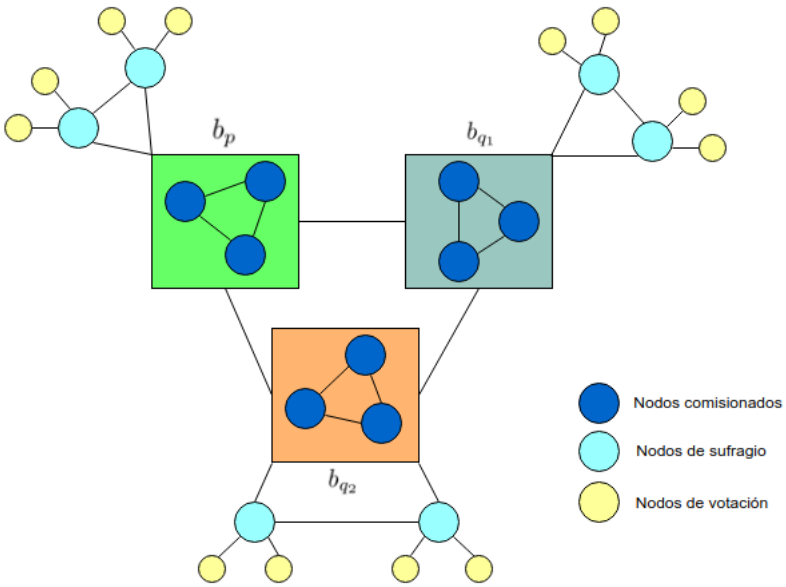

Figura 4.2: Topología de red del blockchain

número de nodos comisionados nnc tiene que cumplir que $nnc \gg b_n$, por razones que se explicarán más adelante, por lo tanto, va a haber más de un nodo comisionado con el mismo blockchain llamados grupos de comisionados que forman el conjunto $GC = \{gc_1, ..., gc_{b_n}\}$. Cada grupo de comisionados contiene nodos que tienen el mismo blockchain y pueden ser definidos de la siguiente manera:

$$gc_j = \{\forall\, nc \,|\, nc \in NC,\, block(nc) = b_j\} \tag{4.1}$$

donde $0 < j <= b_n$. Una de las razones de formar grupos es la redundancia, si es que un nodo comisionado falla o es eliminado de la red por comportamiento inusual, van a existir respaldos del blockchain en la red. Cada grupo de comisionados tiene una fuente de votaciones en donde son guardados los votos para su posterior verificación. Los nodos comisionados son administrados por la organización encargada del proceso de votación, y se tiene la suposición de que se alojan en servidores privados y altamente seguros. Cada nodo comisionado esta virtualmente conectado con todos los nodos comisionados.

Nodos de sufragio

El conjunto de nns nodos de sufragio $NS = \{ns_1, ..., ns_{nns}\}$ contiene los nodos intermediarios entre los nodos votantes y los nodos comisionados. Estos nodos son los encargados de validar los votos y generar bloques con las tuplas $(E_{u_i}, s_j^{v_i})$ de los nodos votantes bajo la supervisión de los nodos comisionados. Al igual que los nodos comisionados, los nodos de sufragio forman un conjunto de grupos $GS = \{gs_1, ..., gs_{b_n}\}$ donde cada grupo de sufragio es obtenido de la misma forma con la Ecuación 4.1. La diferencia está en que los nodos de sufragio sólo guardan el último bloque de la cadena; esto por cuestiones de seguridad, y porque para poder generar nuevos bloques sólo se necesita del último bloque de la cadena.
Un grupo de sufragio gs_i es supervisado por un grupo de comisionados gc_j si cumplen que $block(gs_i) = block(gc_j)$, donde $block()$ retorna el blockchain que guarda cada grupo. Cada nodo de sufragio esta virtualmente conectado con todos los nodos de su grupo y con todos los nodos comisionados del grupo supervisor. Los nodos comisionados guardan un valor de confiabilidad cp_k entre 0 y 1 para cada nodo de sufragio $ns_k \in NS$. Este valor es utilizado en el algoritmo de verificación y consenso para descartar posibles nodos maliciosos.

Nodos votantes

El conjunto de nnv nodos de sufragio $NV = \{nv_1, ..., nv_{nnv}\}$ contiene los nodos más simples de la red. Los nodos votantes son, como su nombre lo indica, los nodos en donde los votantes pueden realizar sus votos. La única funcionalidad de estos nodos es de recolectar y enviar los votos a los nodos de sufragio. Los nodos votantes son los más volátiles del sistema, dependiendo del enfoque. Cuando un nuevo nodo votante se atacha a la red, es conectado a un sólo nodo de sufragio, generalmente el de menor distancia, al cual va enviar los votos realizados en el nodo.

Dependiendo del tipo de sistema electrónica que se quiera implementar, un mismo nodo puede ser de dos tipos diferentes. Por ejemplo, en un sistema no remoto, las máquinas de sufragio tendrían el rol de nodos votantes y de nodos de sufragio. En un sistema híbrido, igualmente las máquinas de sufragio tendría el rol de nodos votantes

y nodos de sufragio y habría nodos enteramente votantes que serían las computadoras personales de los votantes. En un sistema remoto, igualmente habría nodos enteramente votantes y los nodos de sufragio también serían nodos comisionados.

4.3.3. Algoritmo de validación y consenso

El proceso de validación y consenso se da por ciclos de votación y son organizados y supervisados por cada grupo de comisionados. Los nodos votantes envían el conjunto de tuplas S_i' con la credencial electoral y los *shares* del voto cifrado al nodo de sufragio al que esté conectado. El nodo de sufragio envía el conjunto hacia un nodo comisionado. El nodo comisionado que recibe el conjunto guarda de forma aleatoria cada tupla en una fuente de votos de un grupo de comisionados. El primer ciclo de votación de un grupo de comisionados se da cuando la fuente sobrepasa un umbral determinado, y luego se comienza un ciclo inmediatamente después de terminar el anterior. Un ciclo de votación se divide en tres partes: preparación del ciclo, la verificación de votos y la verificación de bloques.

Preparación del ciclo

En el primer ciclo se establecen en 1 los valores de confiabilidad de todos los nodos de sufragio del grupo, y se elige de manera aleatoria un líder comisionado en cada grupo.
Teniendo en cuenta que el ciclo es del grupo de comisionados $\hat{gc}$, en los demás ciclos se sigue los siguientes pasos:

1. El líder comisionado selecciona de manera aleatoria un grupo $gc' \neq \hat{gc}$.
2. El líder comisionado del grupo gc' selecciona de manera aleatoria un nodo del grupo $\hat{gc}$.
3. El nodo seleccionado se vuelve el nuevo líder comisionado del grupo $\hat{gc}$.

Para todos los ciclos, incluido el primer ciclo, el líder comisionado del grupo $\hat{gc}$ selecciona $(|gs|/2)$ nodos de sufragio del grupo gs al que supervisa. Estos nodos son escogidos aleatoriamente en base a una probabilidad. La probabilidad de escoger el nodo i se

calcula con la siguiente fórmula:

$$p_i = \frac{cp_i}{Z} \; donde \; Z = \sum_{i=1}^{|gs|} cp_i$$

donde cp_i es el valor de confiabilidad de el nodo de sufragio i. Los nodos escogidos son los encargados de realizar la verificación de votos y la verificación de bloques.

Verificación de votos

Teniendo el líder comisionado del grupo y el conjunto de nodos de sufragio candidatos escogido, se siguen los siguientes pasos:

1. El líder comisionado obtiene los votos de la fuente de votos y los transmite a cada uno de los nodos candidatos.
2. Para cada voto obtenido de la fuente vf_i se hace:
   a) Cada nodo candidato verifica la validez del voto vf_i y manda su veredicto al líder comisionado. Los veredictos son "Aceptado" o "Rechazado".
   b) El líder comisionado da un lapso de tiempo para recibir los veredictos.
   c) Al finalizar el lapso, por mayoría se decide si se rechaza o se acepta el voto. Si el voto es rechazado, se elimina de todos los nodos, y si es aceptado, es guardado por todos los nodos.
   d) Se actualiza el valor de confiabilidad de los nodos candidato cuyos veredictos hayan diferido de la mayoría con la fórmula: $cp' = cp * \varphi$, donde φ es un valor de tolerancia entre 0 y 1.

Al final se tiene un conjunto de votos válidos.

Verificación de bloques

Teniendo el líder comisionado del grupo, el conjunto de nodos de sufragio candidatos y el conjunto de votos válidos, se siguen los siguientes pasos:

1. Cada nodo candidato genera un bloque con el conjunto de votos válidos y lo envía al líder comisionado.

2. El líder comisionado da un lapso de tiempo para recibir los bloques.

3. Se escoge el bloque con más repeticiones.

4. Se actualiza el valor de confiabilidad de los nodos candidatos cuyos bloques difieran del bloque escogido con la misma fórmula usada en la verificación de votos.

5. El bloque escogido se agrega a la cadena y se actualiza en todos los nodos comisionados del grupo y en todos los nodos de sufragio del grupo supervisado.

6. Se termina el ciclo y se da inicio a uno nuevo.

4.4. Proceso de conteo

Cuando las elecciones terminan, el proceso de votación se cierra y se apagan todos los nodos de sufragio, por consiguiente ningún nodo de votación va a poder conectarse a la red para emitir algún voto. El proceso de conteo se hace en presencia de las autoridades y de las organizaciones encargadas, que hacen entrega de sus *shares* para formar el conjunto SO. El proceso está dividido en dos partes: la regeneración de votos y el conteo de votos.

4.4.1. Regeneración de los votos

Para cada credencial electoral E_{u_i} se recolectan las tuplas de todos los blockchain cuya credencial en la tupla coincida con E_{u_i}. Como resultado se obtiene el conjunto de *shares* S_i. Se reconstruye el voto cifrado E_{v_i} con los conjuntos S_i y SO mediante el modelo de Visual Criptography. El conjunto de votos cifrados es definido por $F = \{E_{v_1}, ..., E_{v_{m'}}\}$ donde m' puede ser un número igual o diferente al número de votantes m.

4.4.2. Conteo de votos

Para el conteo de votos se utiliza la reencriptación con la instancia exponencial del algoritmo ElGamal. La reencriptación es posible ya que ElGamal exponencial es homomórfico. Dado un mensaje cifrado $E = enc_y^+(m, r)$, se obtiene un nuevo valor

aleatorio $r' \in G_q$ y se calcula el mensaje cifrado $E' = reEnc_y^+(E, r') = E \cdot enc_y^+(0, r') = enc_y^+(m, r + r')$. Para hacer un conteo de votos más confiable y con menos escrúpulos, se aplica el mezclado criptográfico dada por $Z' = shuffle_{f_K}^{\phi}(Z)$ (Ver Sección 3.2.3). En donde, para este proceso, se va a tener la función $f_{k_i}(z_i) = reEnc(z_i, k_i)$.
Se obtiene una lista de llaves $K \subset G_q$ de tamaño m' y una permutación aleatoria ϕ. Se aplica un mezclado criptográfico a F y se obtiene $F' = shuflee_{f_K}^{\phi}(F)$. Luego, se descifra el conjunto de votos cifrados obteniendo $H = dec_x^+(F')$, con la llave privada x calculada al inicio del protocolo. Se descartan todos los votos que no se encuentran en el conjunto de identificadores IC y se obtiene el conjunto de votos válidos $H' = H \cap IC$. Finalmente se cuentan las ocurrencias de cada identificador de candidato en H' obteniendo así los resultados finales del proceso de votación.

Capítulo 5

Experimentos y Resultados

5.1. Análisis de Seguridad

5.1.1. Privacidad eterna

Lema 5.1.1. *Se mantiene la privacidad eterna luego del proceso de votación ante un futuro adversario que resuelve de manera eficiente el problema del logaritmo discreto.*

Prueba: Se tiene la suposición que el proceso de votación ha terminado y que en un futuro no muy lejano se descubre una manera eficiente de resolver el problema del logaritmo discreto. Un adversario podría descifrar sin ningún problema la credencial electoral E_{u_i} del votante V_i enlazada al *share* $s_j^{v_i}$ para algún j donde $0 < j \leq b_n$ que se encuentra guardado en el blockchain público b_p. Recordando, estas credenciales tienen la forma $\hat{u_i} = e^{\alpha}e^{\beta}$, donde α y β son parte de la llave privada del votante i. El adversario también podría descifrar la llave pública del votante V_i, $u_i = h_1^{\gamma}h_2^{\delta}$, obteniendo así los valores γ y δ. Con estos valores no hay forma de vincular el *share* $s_j^{v_i}$ con el votante V_i ya que se requeriría de la llave privada $(\alpha, \beta, \gamma, \delta)$ para poder hacer la vinculación, pero esta llave sólo se encuentra en posesión de el votante V_i (y tal vez para ese momento esté perdida); no se encuentra guardada en ningún otro lugar. □

Lema 5.1.2. *Se mantiene la privacidad eterna durante el proceso de votación ante un futuro adversario que resuelve de manera eficiente el problema del logaritmo discreto*

Prueba: Aquí se cumple también la prueba del Lema 5.1.1. Pero en este caso puede ocurrir que un adversario intercepte el conjunto de tuplas S_i' y la llave pública u_i, pertenecientes al votante V_i, mientras son enviadas desde un nodo de votación hacia un nodo de sufragio. El adversario puede enlazar el conjunto S_i' con el votante V_i, pero no podría revelar el voto porque le haría falta el conjunto de *shares* SO de los organizadores encargados. En este caso la privacidad depende de cuan seguro se mantenga el conjunto SO. □

5.1.2. Libre de recibo

Lema 5.1.3. *Un votante no puede probar por quién votó a un coersor simple.*

Prueba: Recordando, luego de que un votante V_i emite su voto, se le hace entrega de una credencial de verificación r_i^u. Lo único que el votante puede hacer con esa credencial es verificar que uno de los *shares* generados a partir de su voto esté en el blockchain público b_p, que significa que su voto ha sido registrado correctamente y está en la red blockchain. Por lo tanto, un coersor no puede obtener información del voto del votante a partir de la credencial de verificación brindada. □

Lema 5.1.4. *Un votante no puede probar por quién votó a un coersor que sea un organizador encargado o esté presente en el proceso de conteo.*

Prueba: Si un coersor tiene la credencial electoral $\hat{u}_i$ y la credencial de verificación r_i^u del votante V_i puede obtener la credencial electoral cifrada E_{u_i}. Pero no puede enlazarlo al voto, porque al momento de la regeneración de los votos, las credenciales electorales son desligadas y descartadas. Además, no podría enlazar la posición en la que aparece la credencial E_{u_i} con la posición en la que aparece el voto cifrado E_{v_i} en el conjunto F ya que este conjunto es mezclado criptográficamente. □

5.1.3. Confiabilidad

En esta parte se va a probar la confiabilidad del algoritmo de consenso.
La probabilidad de escoger un líder comisionado L_i de un grupo comisionado gc_i al inicio del protocolo es:

$$P_0(L_i) = \frac{1}{|gc_i|} \tag{5.1}$$

, donde $|gc_i|$ es el número de nodos comisionado en el grupo gc_i.

La probabilidad de escoger un grupo gc_i se da por el evento G_i y se obtiene como sigue:

$$P(G_i) = \frac{1}{|GC| - 1} \tag{5.2}$$

, donde $|GC|$ es el número de grupos comisionado.

La probabilidad de que un líder comisionado L_j de un grupo comisionado gc_j escoja un grupo comisionado gc_i es un ciclo t esta dada por el evento L_jG_i y se obtiene como sigue:

$$P_t(L_jG_i) = P_{t-1}(L_j) * P(G_i) \tag{5.3}$$

La probabilidad de que un líder comisionado L_i de un grupo comisionado gc_i escoja a un nuevo líder comisionado $L_j^{'}$ de un grupo comisionado gc_j en un ciclo t esta dada por el evento $L_iL_j^{'}$ y se obtiene como sigue:

$$P_t(L_iL_j^{'}) = P_t(L_jG_i) * P_0(L_j^{'}) \tag{5.4}$$

donde L_j es el líder actual del grupo gc_j y $L_j^{'}$ es el nuevo lider escogido del grupo gc_j.

La probabilidad de escoger un nuevo líder comisionado L_j de un grupo comisionado gc_i en un ciclo t es:

$$P_t(L_j) = \frac{\sum_{i=0, i\neq j}^{|GC|} P_t(L_iL_j)}{|GC| - 1} \tag{5.5}$$

La probabilidad un nodo de sufragio ns_i de un grupo de sufragio gs_i es:

$$P(ns_i) = \frac{cp_i}{|gs_i|} \tag{5.6}$$

La probabilidad de que un líder L_i escoja a un nodo de sufragio ns_i esta dada por el evento L_ins_i y se obtiene de la siguiente forma:

$$P_t(L_ins_i) = P_{t-1}(L_i) * P(ns_i) \tag{5.7}$$

La ecuación 5.5 sirve para poder hallar también la probabilidad de que se escoja un líder de grupo maliciosos, si es que el sistema de grupos comisionado ha sido comprometido,

lo cuál es más complicado ya que esta capa de nodos se encuentra fuera del rango de conexiones públicas. La ecuación 5.7 sirve para hallar la probabilidad de que un líder escoja un nodo de sufragio que se encuentra comprometido; esta situación si puede ocurrir con mayor frecuencia, ya que los nodos de sufragio, dependiendo de la implementación, pueden estar ubicados públicamente.
SecIVo [21] es un framework para medir de forma cuantitativa la seguridad en sistemas y protocolos de votación electrónica. SecIvo recibe modelos cualitativos de la seguridad del sistema, como las posibles amenazas a las que se enfrenta el sistema y las fórmulas de probabilidad de que ocurran dichas amenazas.

Lema 5.1.5. *El sistema puede soportar hasta* $\frac{3|gs_i|}{4} + 1$ *nodos de sufragio maliciosos hasta que el grupo* gs_i *sea comprometido por completo.*

Prueba: El líder L_i del grupo de sufragio gs_i escoge $|gs_i|/2$ nodos de sufragio para que participen en la verificación de los votos. Si el atacante tiene el control de $|gs_i|/2$ nodos de sufragio, no le es suficiente, sólo se necesitan $|gs_i|/4+1$ nodos en buen estado para que dominen un ciclo y reduzcan el valor de confiabilidad de todos los demás nodos maliciosos. Entonces, para que el atacante pueda tener control de los ciclos debe infectar al menos $|gs_i|/2+|gs_i|/4+1 = \frac{3|gs_i|}{4}+1$ nodos de sufragio, y así poder controlar a todo el grupo. □

5.2. Implementación del protocolo

El protocolo se implementó simulando un sistema de votación electrónica de tipo remoto. En este caso, los servidores funcionan como nodos comisionado y nodos de sufragio, mientras que las computadoras personales de los votantes funcionan como nodos votantes. La red blockchain fue implementada en el lenguaje Python, mientras que la fase de preparación, votación y conteo de votos fueron implementados en el lenguaje C++.
Una cadena de bloques tiene un archivo de configuración con el bloque que inicia la cadena, el bloque que termina la cadena, el número de bloques que tiene la cadena y el número de votos totales verificados en la cadena. Cada bloque contiene su hash, la lista

de las tuplas con las credenciales y los votos, el hash del bloque padre, y el *timestamp*. La implementación del protocolo sirvió para demostrar su correcto funcionamiento.

Capítulo 6

Conclusiones y trabajo futuro

6.1. Conclusiones

La votación electrónica puede ser una mejor alternativa a la velocidad y a la comodidad de los usuarios, pero cumplir con todos los requerimientos que la constitucionales y de seguridad que la democracia especifica es una tarea ardua. En la presente investigación se desarrolló un protocolo de votación electrónica basado en blockchain que da una solución al problema de las máquinas de votación maliciosas sin eliminarlas totalmente de la red. Además, el protocolo puede ser adaptado a los diferentes tipos de votación electrónica que existe, manteniendo siempre la transparencia y la seguridad de los datos. El protocolo cumple con la privacidad eterna durante y después del proceso de votación, en ningún momento un atacante puede ligar las credenciales del votante con un voto específico. El protocolo también cumple con la libertad de recibos, ningún coersor es capaz de obtener el voto de un votante mediante un dato emitido por el sistema. El modelo de blockchain con el algoritmo de consenso propuestos pueden soportar $3N/4 + 1$ nodos maliciosos, donde N es el número de nodos; mientras que el modelo convencional de blockchain soporta $N/2$ nodos maliciosos.

6.2. Contribuciones

El protocolo propuesto da una solución al problema de las máquinas de votación maliciosas, manteniendo los requerimientos constitucionales y de seguridad que la democracia dicata, tales como la privacidad eterna y la libertad de recibos. Además, el protocolo puede ser adaptado a los diferentes tipos de votación electrónica que existe. El protocolo no cumple, en su totalidad, con la resistencia a la coersión.

6.3. Trabajos futuros

Como trabajos futuros se piensa realizar las implementaciones de todos los tipos de votación electrónica con el protocolo propuesto, y realizar simulaciones a diferentes escalas para evaluar la eficiencia del protocolo y, además, realizar un análisis de seguridad atacando directamente el sistema.

Bibliografía

[1] Z. Zheng, S. Xie, H.-N. Dai, and H. Wang, "Blockchain challenges and opportunities: A survey," *Work Pap.–2016*, 2016.

[2] P. Khandekar, "Visual cryptography," 2013.

[3] S. Kumar and E. Walia, "Analysis of electronic voting system in various countries," *International Journal on Computer Science and Engineering*, vol. 3, no. 5, pp. 1825–1830, 2011.

[4] D. A. Gritzalis, "Principles and requirements for a secure e-voting system," *Computers & Security*, vol. 21, no. 6, pp. 539–556, 2002.

[5] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 4, no. 3, pp. 382–401, 1982.

[6] L. S. Sankar, M. Sindhu, and M. Sethumadhavan, "Survey of consensus protocols on blockchain applications," in *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE, 2017, pp. 1–5.

[7] M. Mursi, M. Salama, and M. Mansour, "Visual cryptography schemes: a comprehensive survey," *Int. J. Emerg. Res. Manage. Technol*, vol. 3, no. 11, 2014.

[8] P. Locher and R. Haenni, "Receipt-free remote electronic elections with everlasting privacy," *Annals of Telecommunications*, vol. 71, no. 7-8, pp. 323–336, 2016.

[9] T. Moura and A. Gomes, "Blockchain voting and its effects on election transparency and voter confidence," in *Proceedings of the 18th Annual International Conference on Digital Government Research*. ACM, 2017, pp. 574–575.

[10] B. Adida, "Helios: Web-based open-audit voting." in *USENIX security symposium*, vol. 17, 2008, pp. 335–348.

[11] C. Burton, C. Culnane, and S. Schneider, "vvote: Verifiable electronic voting in practice," *IEEE Security & Privacy*, vol. 14, no. 4, pp. 64–73, 2016.

[12] S. Delaune, S. Kremer, and M. Ryan, "Coercion-resistance and receipt-freeness in electronic voting," in *Computer Security Foundations Workshop, 2006. 19th IEEE*. IEEE, 2006, pp. 12–pp.

[13] "Elecciones generales 2011, estadísticas del padrón electoral." Jurado Nacional de Elecciones (JNE), 2011.

[14] "Estadísticas de las elecciones generales 2016." Jurado Nacional de Elecciones (JNE), 2016.

[15] (2016) Segunda elección presidencial 2016: Participación ciudadana. Oficina Nacional de Procesos Electorales (ONPE). [Online]. Available: https://www.web.onpe.gob.pe/modElecciones/elecciones/elecciones2016/PRP2V2016/Participacion-ciudadana-Total-Todos-Pie.html

[16] G. S. Grewal, M. D. Ryan, L. Chen, and M. R. Clarkson, "Du-vote: Remote electronic voting with untrusted computers," in *Computer Security Foundations Symposium (CSF), 2015 IEEE 28th*. IEEE, 2015, pp. 155–169.

[17] F. Tiryakioglu, M. S. Kiraz, F. Birinci, and M. Karahan, "Trvote: A new, trustworthy and robust electronic voting system." *IACR Cryptology ePrint Archive*, vol. 2016, p. 331, 2016.

[18] D. F. Aranha, H. Ribeiro, and A. L. O. Paraense, "Crowdsourced integrity verification of election results," *Annals of Telecommunications*, vol. 71, no. 7-8, pp. 287–297, 2016.

[19] K. Gjøsteen and A. S. Lund, "An experiment on the security of the norwegian electronic voting protocol," *Annals of Telecommunications*, vol. 71, no. 7-8, pp. 299–307, 2016.

[20] P. Locher, R. Haenni, and R. E. Koenig, "Coercion-resistant internet voting with everlasting privacy," in *International Conference on Financial Cryptography and Data Security*. Springer, 2016, pp. 161–175.

[21] S. Neumann, M. Volkamer, J. Budurushi, and M. Prandini, "Secivo: a quantitative security evaluation framework for internet voting schemes," *Annals of Telecommunications*, vol. 71, no. 7-8, pp. 337–352, 2016.

[22] R. Hanifatunnisa and B. Rahardjo, "Blockchain based e-voting recording system design," in *Telecommunication Systems Services and Applications (TSSA), 2017 11th International Conference on*. IEEE, 2017, pp. 1–6.

[23] P. McCorry, S. F. Shahandashti, and F. Hao, "A smart contract for boardroom voting with maximum voter privacy," in *International Conference on Financial Cryptography and Data Security*. Springer, 2017, pp. 357–375.

[24] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, "Blockchain," *Business & Information Systems Engineering*, vol. 59, no. 3, pp. 183–187, 2017.

[25] M. Bernhard, J. Benaloh, J. A. Halderman, R. L. Rivest, P. Y. Ryan, P. B. Stark, V. Teague, P. L. Vora, and D. S. Wallach, "Public evidence from secret ballots," in *International Joint Conference on Electronic Voting*. Springer, 2017, pp. 84–109.

[26] M. Matela, "Asymmetric cryptography (asymmcrypto)," Ph.D. dissertation, School of Enginneering, Information and Communications University, 2017.

[27] V. K. Mitali and A. Sharma, "A survey on various cryptography techniques," *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, vol. 3, no. 4, pp. 307–312, 2014.

[28] Y. Desmedt, "Elgamal public key encryption," in *Encyclopedia of Cryptography and Security*. Springer, 2011, pp. 396–396.

[29] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *arXiv preprint arXiv:1704.03578*, 2017.

[30] J. Groth, "Homomorphic trapdoor commitments to group elements." *IACR Cryptology ePrint Archive*, vol. 2009, p. 7, 2009.

[31] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Annual International Cryptology Conference*. Springer, 1991, pp. 129–140.

[32] J. Kurmi and A. Sodhi, "A survey of zero-knowledge proof for authentication," *Int J Adv Res Comput Sci Softw Eng*, vol. 5, no. 1, 2015.

[33] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[34] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[35] D. Schwartz, N. Youngs, A. Britto *et al.*, "The ripple protocol consensus algorithm," *Ripple Labs Inc White Paper*, vol. 5, 2014.

[36] N. Van Saberhagen, "Cryptonote v 2. 0," 2013.

[37] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *Open and Big Data (OBD), International Conference on*. IEEE, 2016, pp. 25–30.

[38] T. Hardjono and N. Smith, "Cloud-based commissioning of constrained devices using permissioned blockchains," in *Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security*. ACM, 2016, pp. 29–36.

[39] M. Sharples and J. Domingue, "The blockchain and kudos: A distributed system for educational record, reputation and reward," in *European Conference on Technology Enhanced Learning*. Springer, 2016, pp. 490–496.

[40] M. Naor and A. Shamir, "Visual cryptography," in *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer, 1994, pp. 1–12.

[41] K. Li, H. Li, H. Hou, K. Li, and Y. Chen, "Proof of vote: A high-performance consensus protocol based on vote mechanism & consortium blockchain," in *High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2017 IEEE 19th International Conference on*. IEEE, 2017, pp. 466–473.