LOKI Vote: A Blockchain-based Coercion Resistant E-voting Protocol

Marwa Chaieb[1] (✉) and Souheib Yousfi[2]

[1] Faculty of Sciences of Tunis, Tunis, Tunisia,
chaiebmarwa.insat@gmail.com
[2] National Institute of Applied Science and Technology, Tunis, Tunisia
souheib.youssfi@gmail.com

Abstract. Creating an online electronic voting system that ensures coercion-resistance and end-to-end verifiability at the same time, has constituted a real challenge for a long period of time. The notion of coercion-resistance was first introduced by Juels, Catalano, and Jakobsson (JCJ) in 2005. Since that time, several research papers have appeared to address the main issue of JCJ scheme (the quadratic complexity of verifying credentials). The majority of these systems have been based on the availability of a secure web bulletin board. Despite this widespread requirement, the notion of an append-only web bulletin board remains vague, and no method of constructing such a bulletin board has been proposed in those papers. Our paper fills the gap and proposes an end-to-end verifiable e-voting protocol based on Blockchain technology. In this research work, we propose a Blockchain-based online electronic voting protocol that ensures all the security requirements expected from secure and democratic elections. Our proposal is inspired from the scheme proposed by Araùjo and Traoré in 2013, which is based on the work of JCJ and has a linear complexity. Called *LOKI Vote*, our scheme is practical for large scale elections and ensures a strong privacy for voters by using a variety of cryptographic primitives. Additionally, our protocol enhance the complexity of the old coercion resistant systems by using a new mix network, called Low Latency Anonymous Routing Protocol, which is characterized by a lower complexity and a higher level of security. Finally, we formally prove the security of LOKI Vote using the automated verification tool, ProVerif, and the Applied Pi-Calculus modeling language.

Key words: Online electronic voting, Coercion-resistance, Blockchain, LOKI, Anonymous credential, Low Latency Anonymous Routing Protocol, Formal security proofs.

1 Introduction

Voting is the backbone of every democratic society. Traditional voting systems suffer from several issues mainly the high cost in both money and time and the lack of transparency and verifiability throughout the voting process. Taking advantages from the proliferation of internet, several online electronic voting protocols have appeared to overcome the limitations of traditional voting systems. Such a system has to ensure an exhausted list of security requirements. This list includes: *Eligibility:* Only eligible and registered voters can participate to the election; *Completeness:* All valid votes are counted correctly; *Soundness:* Invalid votes should be easy to detect and discard; *Robustness:* The protocol can tolerate a certain number of misbehaving voters; *Fairness:* No early results that could influence other voters decisions are made available; *Integrity:* Ballots are not altered or deleted during any step of the election; *Vote-and-go:* A voter does not need to wait for the end of the voting phase or trigger the tallying phase; *Privacy:* It should be impossible to link a vote to a voter without his/her help; *Universal verifiability:* Any interested party should be able to verify the correct computation of the final tally from submitted ballots; *Receipt-freeness:* A voter cannot construct a receipt allowing him/her to prove to

a third party that he voted in a particular way. This would also prevent vote selling; *Coercion-resistance:* Even when a voter interacts with a coercer, the coercer can not be sure of whether the voter obeyed his demand or not.
Designing an online e-voting system that guaranties all the above requirements remains difficult. Indeed, there is always a compromise between end-to-end verifiability and privacy. Coercion-resistance is a strong notion of privacy that has been defined for the first time by Juels, Catalano, and Jakobsson (JCJ) in 2005 [1]. Their proposed system is based on fake and valid anonymous credentials and has a quadratic complexity when tallying votes. Based on this work, several proposals have been appeared to surmount this inherent complexity. These voting systems rely on a public bulletin board (PBB), where they post votes and other public parameters, without however specifying how this can be implemented. They make the assumption that this public bulletin board ensures the end-to-end verifiability, fairness, and correctness of the election process. Thus, public bulletin boards must have the following properties: (1) *Distributed architecture* to withstand Distributed Denial Of Service (DDOS) attacks, (2) *Time stamped* to reference data by their dates of publication, (3) *Immutable* to ensure resistance against adding, removing or altering posted data and finally (4) *Universally verifiable* to ensure a high level of transparency. These are exactly the main characteristics of Blockchain technology. Blockchain is a distributed ledger that operates without the need to a trusted party. It can be seen as a digital, decentralized, public and large register where all exchanges made between its users are recorded in a public and secure way. In this paper, we propose a coercion resistant Blockchain-based online electronic voting protocol, called *LOKI Vote*.

Contributions: Our contributions can be summarized as follow:

- Based on the work of Araùjo and Traoré [2], we design an online electronic voting protocol that satisfies the above security requirements and has a linear complexity when tallying votes,
- Called *LOKI Vote*, our proposed system is based on Blockchain technology to ensure end-to-end verifiability and integrity of the election process,
- *LOKI Vote* is designed to be implemented over Loki[1] platform. This Blockchain-based platform comes with a novel mix network, called Low Latency Anonymous Routing Protocol (LLARP), that has a lower complexity than some existing mix networks and fix their vulnerabilities,
- Finally, we formally evaluate the security of the protocol, using ProVerif and Applied Pi-Calculus.

Paper organization: Our paper is organized as follow: in the next section, we review some of the existing coercion resistant schemes. Section3 presents the cryptographic primitives and technologies used in our protocol. Section4 is a detailed description of *LOKI Vote* and its different stakeholders and phases. We discuss the security of our proposed scheme in Section5 and finally Section6 is dedicated to the conclusion and a set of perspectives.

2 Related Work

In this section, we give an overview of some e-voting schemes that are, or claimed to be, coercion resistant. We start by describing three protocols from the literature that are interesting for our work and did not use Blockchain technology (1, 2 and 3). Then, we present two online e-voting systems based on Blockchain technology and claimed to be coercion resistant (4 and 5). We evaluate the security of these systems in Table 2.

[1] https://loki.network/wp-content/uploads/2018/10/LokiWhitepaperV3_1.pdf

1. **Coercion Resistant Electronic Elections (CREE) [1]:** In their paper, Juels, Catalano, and Jakobsson (JCJ) give the first formal definition for coercion-resistance and propose the first coercion resistant e-voting system. Their scheme relies on a secret random string "σ" that serves as an anonymous credential for eligible voters. Each eligible voter gets a valid anonymous credential during the registration phase, after verifying his/her eligibility by an authority called Registrar (R). To vote, each voter encrypts his/her anonymous credential, using a modified version of El-Gamal cryptosystem, and sends it with his/her ballot to a public bulletin board (PBB). Authors make the assumption that the PBB is universally accessible, to which every party can write and read data but no one can alter or delete information from it. After the end of the voting phase, an authority called Talliers (T) perform a blind comparison (using Plaintext Equivalence Test PET [3, 4]) between hidden credentials and a list L of encrypted credentials published by R alongside the plaintext names of registered voters. The list of hidden credentials and L are passed through a re-encryption mix network [5, 6] before being compared to each other. T retain only votes that their corresponding credentials match an element of L, according to PET. Finally, T decrypt all eligible valid votes and tallies the final result.
   The JCJ scheme ensures coercion-resistance thanks to the use of anonymous credentials σ. Indeed, when a voter V_i is under coercion, he/she can simply select and reveal a random group element σ'_i, claiming that this is the credential σ_i. As the coercer is unable to distinguish between a valid credential and a fake one, he can not be sure if the coerced voter obeyed to his demand or not.
   The main drawback of JCJ's scheme is its quadratic complexity in the number of voters during the tallying phase (when verifying the validity of credentials). This issue makes the scheme unrealistic since it can not be employed in a real-world context. Even so, the protocol is widely discussed and taken as a starting point for further improvements [7, 8, 9, 10, 2, 11].
2. **Towards Practical and Secure Coercion Resistant Electronic Elections (TPSCREE) [9]:** To overcome the drawbacks of JCJ scheme, authors propose a new coercion resistant election approach with linear complexity. This solution relies on the BBS group signature scheme [12]. In their paper, authors first describe an attack on Schweisgut scheme [13] (which is also based on the work of JCJ) and prove that it is not coercion resistant as claimed since a coercer can verify later if the coerced voter obeyed to his demand and gave him a valid credential or not. Then, they propose their voting scheme and prove, formally, that is coercion resistant and suitable for large scale elections. The proposed protocol is based on the same cryptographic primitives as JCJ proposal, namely: a public bulletin board [14], the modified El-Gamal cryptosystem proposed by JCJ [1], a universally verifiable mixnet [6, 15], a set of zero knowledge proofs [16, 17] and PET [3]. It unfolds in the following stages: *Registration Phase:* the registrars verify the eligibility of every voter and provides him/her by a credential that has the following form (A,r,x) where $A = (g_1 g_3^x)^{1/(y+r)}$, g_1 and g_3 are public parameters, x is a secret value, y is the private key of R and r is a random value; *Voting Phase:* each voter encrypts his/her vote and credential and casts them via a PBB, including with them a set of proofs to justify the validity of the voting tuple; *Tallying Phase:* the talliers record voting tuples from the PBB, verify the validity of each one, eliminate duplicates and tuples with invalid credentials, then decrypt the remaining votes and count the election final result. When under coercion, a voter gives a fake credential to the coercer. A fake credential has the following form (A,r,x') where $x \neq x'$.
   This protocol presents two main issues. (1) A set of malicious registrars have the possibility to provide ineligible voters by valid credentials. Thus, the final tally may include valid but illegitimate votes. (2) It is impossible to run another election, that has a different list

of eligible voters from the first one, without performing the registration phase another time because authorities do not have the possibility to revoke credentials that are no more eligible.

3. **A Practical Coercion Resistant Voting Scheme Revisited (PCRVSR) [2]:** In 2013, R. Araùjo and J.Traoré pointed out the drawbacks of the previous scheme [9] and propose a revisited version to overcome these issues. They add some modifications in the election process to make the verification of votes eligibility and credential revocation possible. To resolve the first issue, the registrars construct and publish a list L_2, during the registration phase, that contains $< E_T[A], ID_{voter} >$ for each registered voter, where T is the public key of the talliers. During the tallying phase, talliers compare valid credentials in the voting tuples with the list L_2 and count only votes that their credentials match an element from L_2. To resolve the second one, the registrars generate for each new election new key pair and use it to generate new credentials. They calculate the new credentials from the new private key and a list L_1 that retains the couple $< E_R[g_1 g_3^x], ID_{voter} >$ for each voter. The list L_1 is published on the PBB during the first election. The new credentials have the following form $(A' = (g_1 g_3^x)^{1/(y'+r')}, r', x)$, where r' is a random number, y' is the new secret key of the Registrars and x is the same secret value given to the voter during the first time registration.
4. **Platform-Independent Secure Blockchain-based Voting System (PISBVS) [18]:** It is an independent e-voting system implemented on a Byzantine Fault Tolerance consensus [19] based Blockchain. Authors claim that their solution does not rely on a centralized trusted party to compute and publish the election final result, but they still need to trust an administrator to decrypt the sum of votes and upload the result to the Blockchain. They use Paillier cryptosystem to encrypt votes before publishing them on the election Blockchain. It recalls proof of knowledge to ensure correctness and consistence of votes, and Short Linkable Ring Signature (SLRS) to guarantee voters privacy. However, this protocol does not ensure voters eligibility since a voter can register him/herself by simply providing his/her e-mail address, ID number or an invitation URL with a password and these mechanisms are not sufficient to verify the eligibility of a voter. In addition, authors claim to ensure coercion-resistance under the following assumption "*it is assumed that no one stand behind a voter or uses digital devices to record the voting process. We do not take the physical voting environment security into our consideration*". Thus, referring to the definition of coercion-resistance given by JCJ [1], this protocol is not coercion resistant. A coercer can vote in the place of a voter if he knows the voter's secret key. The coerced voter cannot provide a fake secret key to the coercer because a vote with a fake secret key is rejected by the voting smart contract.
5. **Efficient, Coercion-free and Universally Verifiable Blockchain-based Voting (ECFU-VBV) [20]:** It is a Blockchain-based e-voting protocol, claimed to be secure and coercion-resistance without the need to use valid and fake credentials. It uses a randomizer token, a tamper resistant device that can be instantiated with smart cards or Trusted Platform Module (TPM) [21] enabled devices. Authors use Bitcoin to ensure verifiability. Its tallying phase has a linear complexity. It unfolds in the following phases: *Setup:* the election authority generates its public and private keys along with other system parameters; *Register:* a voter V_i interacts with the registrar R to get a pair of public/private keys along with a signed commitment C_i on values s_i, r_i generated by V_i's token randomizer. The voter's credential is the signed version of the commitment using the voter's private key; *Vote:* each voter encrypts its choice v using a one-time key K_i. Then, he/she computes a proof π_i to prove knowledge of r_i using zero-knowledge Succinct Non-interactive ARguments of Knowledge (zk-SNARKs) [22]. He/She casts a tuple that has the following form $< \pi_i, s_i, E_{K_i}(v) >$ via the election Blockchain; *Tally:* the election authority checks the validity of each ballot posted on the blockchain using π_i and eliminates ballots with invalid proofs. It also eliminates duplicates using the element s_i. Then, it decrypts votes using K_i, which are published by voters along-

side with the value s_i to facilitate matching the key to her previously transmitted encrypted vote, and computes the final result. In this paper, authors suppose that the coercer and the voter are not side-by-side. All that the attacker can do is to issue instructions and ask for proof of compliance. Accordingly to the definition of coercion-resistance of JCJ [1], plus the fact that the voter can vote only once, this scheme is not coercion resistant.

3 Basic Notions

In this section, we give an overview of the main cryptographic primitives and technologies used in our protocol.

3.1 El-Gamal cryptosystem

The proposed protocol uses a threshold version of El-Gamal cryptosystem proposed by JCJ in [1]. In this scheme, the key pair is constructed by cooperation between n authorities. It requires t out of n authorities to decrypt a ciphertext. As proved in [1], this modified version of El-Gamal is semantically secure under the Decision Diffie Hellman (DDH) assumption [23]. This variant can be described by the following steps:

- **Key Generation:** Let $\mathbb{G}$ be a cyclic group of order a prime number q, in which the DDH assumption holds. We denote the public key by y and it is represented by the following tuple: $y = (g_1, g_2, h)$; where $h = g_1^{x_1} g_2^{x_2}$. Its corresponding private key is the couple (x_1, x_2); where $x_1, x_2 \in \mathbb{Z}_q$.
- **Encryption:** The ciphertext of a message $m \in \mathbb{G}$ is represented by the following tuple: $E_y[m] = (\alpha, \beta, \gamma) = (m \cdot h^r, g_1^r, g_2^r)$; Where r is a random number from $\mathbb{Z}_q$.
- **Decryption:** m is obtained from (α, β, γ) using the following formula: $m = \alpha / (\beta^{x_1} \gamma^{x_2})$

3.2 Proof of knowledge

Our protocol recalls the Non-Interactive Zero Knowledge Proof (NI-ZKP)[24] during the voting phase to prove the validity of the tuple formed by the voter.

Zero Knowledge Proofs (ZKP)[25] are cryptographic primitives that allow one party, called "prover", to prove to another party, called "verifier", that he knows a secret without revealing the secret itself or any additional secrets. NI-ZKP[26, 27] is a variant of ZKP in which no bidirectional interaction between the prover and the verifier is needed .

3.3 Group Signature Scheme of Boneh, Boyen and Shacham

In their paper [12], Boneh, Boyen and Shacham presented a short group signature scheme. Its security relies on the Strong Diffie-Hellman (SDH) [28] and Decision Linear (DL) [12] assumptions.

Our proposed e-voting protocol uses the BBS group signature scheme, presented in Section 8 of the paper [12], as anonymous credentials for eligible voters. This scheme can be described as follow: Let $\mathbb{G}$ be a cyclic group of order a prime number q, in which the DDH assumption holds, g_1, $g_2 \in \mathbb{G}$ are two random generators, y is a secret key, and $r, x \in \mathbb{Z}_q$ are two random numbers. The signature is represented by the tuple (A, r, x) where $A = (g_1 g_2^x)^{1/(y+r)}$.

3.4 Loki

Loki [2] is a platform based on Monero [3] Blockchain. It proposes significant modifications on Monero source code to ensure a high degree of privacy and provide a model for anonymous

[2] https://loki.network/wp-content/uploads/2018/10/LokiWhitepaperV3_1.pdf

[3] https://www.allcryptowhitepapers.com/wp-content/uploads/2018/05/monero-whitepaper.pdf

transactions and decentralized communication. The main drawbacks of Monero Blockchain are the significant bandwidth and disk space that its node operators require plus the fact that they are not rewarded for their work. To fix this problem, Loki comes with a novel node reward scheme that provides economic incentives for node operators, called *Service Nodes*. These service nodes ensure the privacy and the security of the network. This technology has been proposed to provide internet neutrality, digital anonymity and censorship-resistant suite of tools allowing people to communicate in a private and secure way. This is why Loki can be used in various areas especially when we need to ensure a high level of privacy and anonymity, such as in e-voting systems.

Loki recalls several cryptographic primitives namely *Ring Signature [29]* to obfuscate the true history of transaction outputs, *Stealth Address [30]* to ensure the unlinkability between the receiver true public key and his transactions and *Ring Confidential Transactions [31]* to obfuscate transaction amounts. This Blockchain-based platform also uses the proof of work consensus algorithm to validate transactions and construct blocks. It opts for a different way of block reward distribution: 45% of the block reward are reserved for miner, 50% for service node and 5% for governance operations. The main role of service nodes is to operate the Low Latency Anonymous Routing Protocol[4], which is an anonymous mixnet, and form the Lokinet, which is a fully decentralized network that does not rely on any trusted authority. The Low Latency Anonymous Routing Protocol (LLARP) is a private routing layer created by Loki. It is an hybrid between The Onion Routing (TOR)[5] and Invisible Internet Protocol (I2P)[6]. It fixes vulnerabilities of TOR and I2P protocols and provides a higher level of security and distribution than any existing routing protocol. To better understand how LLARP works, we recall TOR and I2P protocols. The advantages and disadvantages of each protocol are summarized in Table 1.

	The Onion Routing (TOR)	**Invisible Internet Protocol (I2P)**
Advantages	+ Provides an anonymous network, + Preserves internet privacy, + Performs better at evading state-level firewalls, + Ensures a high level of censorship resistance.	+ Provides an anonymous network, + Uses a Distributed Hashing Table (DHT) instead of directory authorities, + Allows both TCP and UDP traffics.
Disadvantages	- It is an hierarchical network, - Relies on a group of directory authorities (centralized servers), - Trusting claimed capacity, - Allows only TCP traffic, - Irresistant to Sybil attacks.	- Problems of performance and lack of bandwidth, - Tunnels are short lived, - Irresistant to Sybil attacks.

Table 1: Advantages and disadvantages of TOR and I2P

Both TOR and I2P are operated by volunteers, which can cause problems of security, reliability and performance. In fact, a network constructed from financial incentives can achieve a greater resilience against attacks, while providing a more reliable service. This is what proposes the LLARP by using a Distributed Hashing Table (DHT) based on Blockchain technology. This Blockchain-based DHT allows service nodes to act as routers in the network and they are rewarded for their work. LLARP also opts for packet switched based routing instead of tunnel

[4] https://github.com/loki-project/loki-network
[5] https://www.torproject.org/
[6] https://geti2p.net/en/

based-routing to allow better load balancing and redundancy in the network. To avoid Sybil attacks, LLARP allows only service nodes to route packets, and they are rewarded for their honesty.

4 Protocol Description

We propose a coercion resistant online e-voting system that uses Blockchain technology and designed to be implemented over Loki. Called *LOKI Vote*, our protocol provides an end-to-end verifiability by using a Blockchain-based public bulletin board to display all public values and offer a persistent view to all voters. In this section, we present the different entities involved in *LOKI Vote* as well as its different phases.

4.1 Entities

Our protocol involves three main entities:

- *Registration authorities (RAs)*: They cooperate and generate a new key pair (R, R') for each new election, generate and publish the election parameters on the Blockchain during the setup phase, verify the eligibility of every person wishing to register to the election, during the registration phase, and provide only eligible voters by anonymous credentials which are constructed by cooperation between all RAs. In addition, they cooperate to construct and publish two lists L_1 and L_2 which serve later, respectively, for credential revocation and verification of votes eligibility. Finally, they help the tallying authorities to verify the validity of credentials during the tallying phase.
- *Tallying authorities (TAs)*: They cooperate and generate a key pair $(\mathscr{T}, \mathscr{T}')$ during the setup phase, read voting tuples from the election Blockchain, verify, decrypt and compute eligible and valid votes during the tallying phase. Finally, they publish the final tally on the Blockchain.
- *Eligible voters (V)*: Every eligible voter (V_i) has a unique valid credential per election to vote with, and can generate an unlimited number of fake credentials to use them when he/she is under coercion. He/she has the right to vote more than once before the end of the voting phase and only his/her last and valid vote is counted.

Every entity in our protocol has a read and write access to our election Blockchain, which is considered as a public bulletin board and ballot box. Also, observers and election organizers have the right to access the Blockchain and supervise the election to ensure the correctness of the election process.

4.2 Phases

Our protocol unfolds in four phases: setup, registration, vote and tally. There are two ways to perform the setup and the registration phases, depending on whether it is the first time the protocol is runned (the first election) or more.

Setup Phase

Setup for the first election: This phase is described by Figure 1.

1. RAs start by generating the following election parameters and publish them on the election Blockchain: $\mathbb{G}$ a cyclic group of order a prime number q, in which the Decision Diffie Hellman problem holds; g_1, g_2, g_3 and $o \in \mathbb{G}$ four random generators. They also cooperate and generate their key pair (R, R'), where $R = g_3^y$ is the public key and $R' = y$ is the private one. A Modified El-Gamal threshold [1] key pair $(\mathscr{R}, \mathscr{R}')$ is also generated by cooperation between all RAs. Finally, they publish the public parts on the election Blockchain.

2. TAs cooperate and generate a key pair of Modified El-Gamal threshold $(\mathscr{T}, \mathscr{T}')$, where $\mathscr{T} = (g_1, g_2, h = g_1^{x_1} g_2^{x_2})$ is the public part and $\mathscr{T}' = (x_1, x_2)$ is the secret one. They publish their public key on our Blockchain.

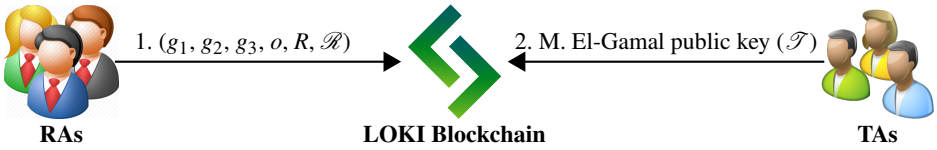

Fig. 1: Setup Phase, First Election

Setup for the second (or more) election: For each new election, RAs create a new random generator $o' \in \mathbb{G}$. If we have no need to revoke the old credentials, RAs publish the same election parameters as the first ones, with replacing o by o' (Figure 2).

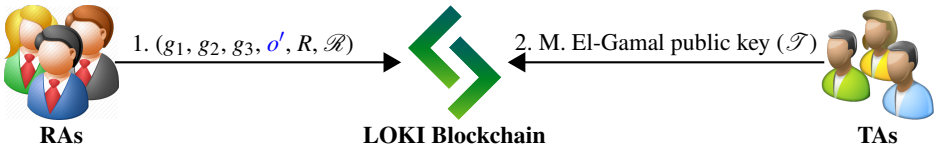

Fig. 2: Setup Phase, Second or more Election, Without Revocation

Otherwise, they generate a new key pair (R_1, R_1'), where $R_1 = g_3^{y_1}$ and $R_1' = y_1$ and publish all public parameters on the election Blockchain (Figure 3). The new key pair is used for credential revocation.

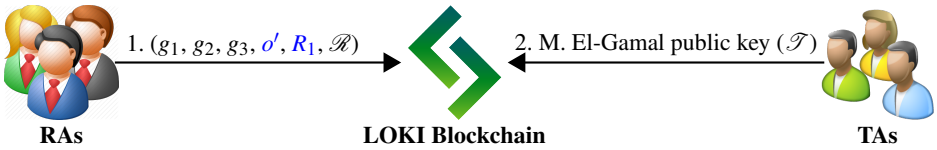

Fig. 3: Setup Phase, Second or more Election, With Revocation

Registration Phase

Registration for the first election: Every person who has the right to vote and wishes to do so, physically moves to the nearest polling station and provides his/her identity card to the registration authorities (RAs). These authorities verify his/her eligibility and provides him/her by a valid and anonymous credential if he/she is eligible to participate to the election. Otherwise, the registration phase fails. Figure 4 illustrates a successful registration phase. The credential is calculated by cooperation between the registration authorities and is used by the voter to cast a vote during the voting phase. To calculate the credential, RAs generate two random numbers $r, x \in \mathbb{Z}_q$, use their shared private key y and calculate $A = (g_1 g_3^x)^{1/(y+r)}$. The credential is formed by the tuple (A, r, x) where x is the secret part of the credential. After registering all eligible voters, RAs cooperate and generate two lists:

- $L_1 =< E_{\mathscr{R}}[g_1 g_3^x], ID_{voter} >$ contains, for each voter, the ciphertext of $(g_1 g_3^x)$ using their public key $\mathscr{R}$ with the corresponding unique voter identifier ID_{voter}. This list will serve later for credential revocation.

– $L2 =< E_{\mathscr{T}}[A], ID_{voter} >$ contains, for each voter, the ciphertext of A using TAs public key $\mathscr{T}$ with the corresponding unique voter identifier ID_{voter}. This list serves for verification of credentials eligibility.

Finally, RAs publish L_1 and L_2 on the election Blockchain.

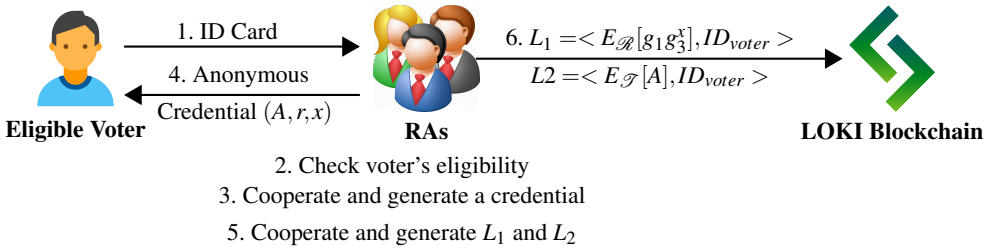

Fig. 4: Registration Phase, First Election.

Registration for the second (or more) election: For each new election, and if there is one or more credentials to revoke, RAs need to update credentials for voters who still have the right to vote. From the list L_1 and their new shared private key y_1, they calculate the new valid anonymous credentials. By inspecting the values ID_{voter}, the RAs identify voters that can vote in the new election. For each of these voters, RAs choose randomly $r_1 \in \mathbb{Z}_q$ and calculate his/her new valid credential $\sigma_1 = (A_1, r_1, x)$, where $A_1 = (g_1 g_3^x)^{1/(y_1+r_1)}$ and x is the same secret value given to the voter during his/her first time registration. At the end of this phase, RAs publish on the election Blockchain the lists $L_3 =< (A_1, r_1), ID_{voter} >$ and $L_4 =< E_{\mathscr{T}}[A_1], ID_{voter} >$. This phase is illustrated by Figure 5.

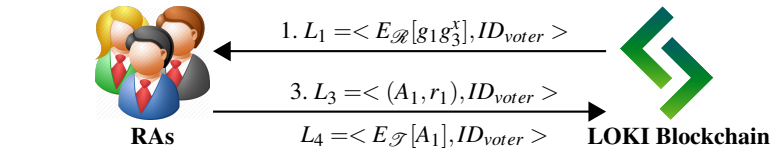

Fig. 5: Registration Phase, Second or more Election

Voting Phase To cast a vote, each eligible voter constructs a voting tuple that contains his/her encrypted vote, his/her encrypted credential and a set of Non-Interactive Zero Knowledge Proofs that prove the correctness of the tuple. It has the following form:

$$< E_{\mathscr{T}}[V], E_{\mathscr{T}}[A], E_{\mathscr{T}}[A^r], E_{\mathscr{T}}[g_3^x], o^x, \mathscr{P} >$$

Where $\mathscr{T}$ is the public key of TAs, V is the choice of the voter, A, r, and x constitute the voter's credential and $\mathscr{P}$ is composed of a set of NI-ZKP. These proofs are constructed by using standard techniques such as [16] and contain: P_1*:* Proof of validity of the encrypted vote V; P_2*:* Proof of knowledge of the plain-text related to $E_{\mathscr{T}}[A]$; P_3*:* Proof of knowledge of the plain-text related to $E_{\mathscr{T}}[A^r]$; P_4*:* Proof of knowledge of the plain-text related to $E_{\mathscr{T}}[g_3^x]$; P_5*:* Proof related to the value of A to ensure that is different from 1; P_6*:* Proof of knowledge of the discrete logarithm of o^x in the basis o and its equality to the discrete logarithm of the plain-text related to $E_{\mathscr{T}}[g_3^x]$ in the basis g_3. This phase is illustrated by Figure 6. The voter has the right to cast more than one tuple before the end of the voting phase and only his/her last valid vote is counted. When he/she is under coercion, the voter generates $x' \neq x$ and constructs a tuple using the value of x' instead of x. If it is not the first election, the voter uses o' instead of o and his/her new valid credentials σ_1 that he/she received from the RAs during the registration phase.

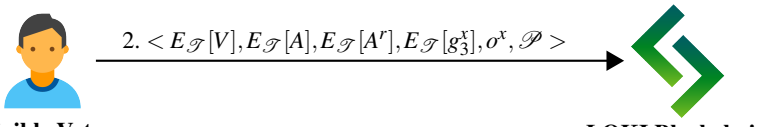

Fig. 6: Voting Phase

Tallying Phase After the end of the voting phase, the tallying authorities read all voting tuples from our election Blockchain and proceed to the tallying process. They start by checking the validity of every tuple proofs and discard the ones with invalid proofs. Then, they eliminate duplicates using the attribute o^x (or o'^x if it is not the first election) included in each tuple, using a hash table. As all voting tuples were sent through LOKI network, they have been passed through the LLARP mix network (see section3.4 for more details). At this step, each voting tuple has the following form: $< E'_{\mathscr{T}}[V], E'_{\mathscr{T}}[A], E'_{\mathscr{T}}[A^r], E'_{\mathscr{T}}[g_3^x] >$. Using the three last elements of each tuple, TAs cooperate with RAs and check the validity of the anonymous credentials. They proceed as follow:

- Using their shared secret key y, RAs cooperate and calculate $E'_{\mathscr{T}}[A]^y$ which is equal to $E'_{\mathscr{T}}[A^y]$ thanks to El-Gamal homomorphic property. Then, they perform the following multiplication: $E'_{\mathscr{T}}[A^y] \cdot E'_{\mathscr{T}}[A^r] = E'_{\mathscr{T}}[A^y \cdot A^r] = E'_{\mathscr{T}}[A^{y+r}]$. The first equality is obtained by using the homomorphic property of El-Gamal cryptosystem.
- TAs cooperate and perform the following multiplication, in which they also use the homomorphic property of El-Gamal: $E'_{\mathscr{T}}[A^{y+r}] \cdot E'_{\mathscr{T}}[g_1]^{-1} \cdot E'_{\mathscr{T}}[g_3^x]^{-1} = E'_{\mathscr{T}}[A^{y+r} \cdot g_1^{-1} \cdot g_3^{-x}]$. The result $E'_{\mathscr{T}}[A^{y+r} \cdot g_1^{-1} \cdot g_3^{-x}]$ is denoted C. Then, TAs execute the PET to determine whether C is an encryption of 1 or not. If it is the case, the credential is judged valid and the corresponding tuple passes to the next step. Indeed, a valid credential has the following form $\sigma = (A, r, x)$ where $A = (g_1 \cdot g_3^x)^{1/(y+r)}$ so we have $A^{y+r} = g_1 \cdot g_3^x$ thus $A^{y+r} \cdot g_1^{-1} \cdot g_3^{-x} = 1$. Otherwise, the credential is judged invalid and the voting tuple is discarded.

The next step consists on verifying the eligibility of votes by using the element $E'_{\mathscr{T}}[A]$ included on each voting tuple and the list L_2. We recall that $L_2 =< E_{\mathscr{T}}[A], ID_{voter} >$ was published on the election Blockchain by RAs during the registration phase. At this step, and after being passed through the LLARP mix network, we obtain $L'_2 =< E'_{\mathscr{T}}[A], ID'_{voter} >$. By using a hash table, TAs compare $E'_{\mathscr{T}}[A]$ coming on each voting tuple to each $E'_{\mathscr{T}}[A]$ included on the list L'_2 and maintain only tuples that match an element from L'_2. Finally, TAs cooperate and decrypt all votes of the retained list, using their shared secret key $\mathscr{T}'$, and compute the election final result.
We mention that if it is not the first election, y is replaced by y_1, A and r are replaced, respectively, by A_1 and r_1 and L_2 by L_4.

5 Security Evaluation

In this section, we discuss, formally and informally, the security of our proposed scheme.

5.1 Informal Security Evaluation

We start by evaluating our protocol against the list of security requirements presented in the Introduction section. We resume this evaluation in Table 2.

- Eligibility: *LOKI Vote* includes a face to face registration phase, in which the RAs verify the eligibility of every voter and provides only eligible ones by valid credentials. At the end of this phase, RAs publish the list L_2 of all registered voters. Thus, everyone can verify the validity of this list. In addition, during the tallying phase, TAs count only votes that match an element from L_2.

- Completeness: TAs ensure that all valid votes are counted correctly and give proofs for the correctness of their work.
- Soundness: This property is ensured by using the set of proofs included in each voting tuple. Indeed, TAs discard all tuple with invalid proofs from the final tally.
- Robustness: Our proposed protocol is resistant to the misbehavior of malicious voters.
- Fairness: All votes are encrypted, using the TAs public key $\mathscr{T}$, before being cast. Thus, no one, except TAs, has the possibility to decrypt votes and get partial results before the official tally. We mention here that the decryption private key is constructed by cooperation between all TAs. So, we need to trust only one TA to ensure fairness.
- Integrity: The fact of casting and storing votes and the other voting data in the Blockchain safeguard them from being altered or deleted thanks to the immutability property of Blockchain technology.
- Vote-and-go: *LOKI Vote* does not need the voter neither to wait for the end of the voting phase nor to trigger the tallying one. He can simply cast a vote and quiet the voting system.
- Privacy: This property is ensured by using the Loki platform, which is built on the top of Monero Blockchain. Monero is characterized by the anonymity of its transactions since it uses ring signature and ring confidential transactions primitives. Thus, we can not link a transaction to its sender. Consequently, we can not link a voter to his/her vote.
- Universal verifiability: This property in ensured by using Blockchain technology as a public bulletin board. Except the registration phase, all our protocol phases are on chain. Thus, voters, election organizers, observers and any interested party have the possibility to watch the voting process and verify the correctness of each step as well as the final tally.
- Receipt-freeness: From all public data, which are written on the election Blockchain, the voter can not construct a receipt that reflects his/her vote.
- Coercion-resistance: *LOKI Vote* is inspired from the scheme [2], which is formally proved coercion resistant. This property is ensured by using the BBS signature scheme $\sigma = (A, r, x)$ as anonymous credentials for eligible voters. When they are under coercion, voters disclose a random value x' instead of x and pretend that $\sigma' = (A, r, x')$ is the valid credential. Since the voter has the right to vote more than once, he/she has the possibility to cast another vote when he/she is lonely and uses his/her valid credential. The coercer has no possible way to verify if the voter obeyed to his instructions or not.

	CREE	**TPSCREE**	**PCRVSR**	**PISBVS**	**ECFUVBV**	**LOKI Vote**
Eligibility	✗	✗	✓	✗	✓	✓
Completeness	✓	✓	✓	✓	✓	✓
Soundness	✓	✓	✓	✓	✓	✓
Robustness	✗	✗	✗	✓	✓	✓
Fairness	✓	✓	✓	✓	✓	✓
Integrity	✗	✗	✗	✓	✓	✓
Vote-and-go	✓	✓	✓	✓	✓	✓
Privacy	✓	✓	✓	✓	✓	✓
Universal verifiability	✓	✓	✓	✓	✓	✓
Receipt-freeness	✓	✓	✓	✓	✓	✓
Coercion-resistance	✓	✓	✓	✗	✗	✓

Table 2: Security Evaluation of CREE, TPSCREE, PCRVSR, PISBVS, ECFUVBV and *LOKI Vote*

5.2 Formal Security Evaluation

In this part, we perform an automated security analysis using the verification tool ProVerif [32]. It is an automatic symbolic protocol verifier, capable of proving *reachability properties*, *correspondence assertions*, and *observational equivalence* [33] of a given protocol described in Applied Pi-Calculus [34]. This modeling language is a variant of the Pi-Calculus extended with equational theory over terms and functions and provides an intuitive syntax for studying concurrency and process interaction. The Applied Pi-Calculus allows us to describe several security goals and to determine whether the protocol meets these goals or not. We use the classical intruder model and the standard modeling of the security properties proposed by Dreier et al. [35] in our ProVerif code.

Because of the limitation on the number of pages, we put all ProVerif codes online[7]. We define the following queries to prove votes secrecy, voters' authentication and votes privacy and give the results of executing the codes, and the time it takes ProVerif to prove the properties in Table 3.

- **Verification of votes secrecy:** To capture the value of a given vote, an attacker has to intercept the values of the parameter *Vote*. Thus we use the following query:

```
query attacker(Vote)
```

- **Verification of voters authentication:** Authentication is captured using correspondence assertions. The protocol is intended to ensure that the TAs verify the eligibility of all voters by verifying the validity of their credentials. Therefore, we define the following events and query:

```
event ValidCred.
event CredentialVerification.
query event(ValidCred)==>event(CredentialVerification).
```

- **Verification of votes privacy:** To express votes privacy we prove the observational equivalence property between two instances of our process that differ only in the choice of votes. To do that, we use `choice[V1,V2]` to represent the terms that differ between the two instances. Likewise, we use the keyword `sync` to express synchronization which help proving equivalences with choice since they allow swapping data between processes at the synchronization points.

Properties	Result	Time
Vote secrecy	Proved	0.007 s
Voter authentication	Proved	0.009 s
Vote privacy	Proved	0.089 s

Table 3: ProVerif results and execution times.

6 Conclusion

We have proposed an end-to-end verifiable, coercion resistant and secure Blockchain-based online e-voting protocol. LOKI Vote is based on the work of Araùjo and Traoré [2] and uses Loki platform. It recalls several cryptographic primitives namely NI-ZKP, Modified El-Gamal, BBS signature and LLARP mix network. It has a linear complexity which makes it practical for large scale elections. We have also proved, formally by using ProVerif, the security of our protocol. Future work will be devoted to implement and evaluate the performance and scalability of the proposed protocol.

[7] `https://drive.google.com/drive/folders/1rJRUAuOdnRHLo40umY6Lq9CRrYwZBLw3?usp=sharing`

References

1. Juels, A., Catalano, D., Jakobsson, M.: Coercion-resistant electronic elections. In Atluri, V., di Vimercati, S.D.C., Dingledine, R., eds.: Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society, WPES 2005, Alexandria, VA, USA, November 7, 2005, ACM (2005) 61–70
2. Araújo, R., Traoré, J.: A practical coercion resistant voting scheme revisited. In Heather, J., Schneider, S.A., Teague, V., eds.: E-Voting and Identify - 4th International Conference, Vote-ID 2013, Guildford, UK, July 17-19, 2013. Proceedings. Volume 7985 of Lecture Notes in Computer Science., Springer (2013) 193–209
3. Jakobsson, M., Juels, A.: Mix and match: Secure function evaluation via ciphertexts. In Okamoto, T., ed.: Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings. Volume 1976 of Lecture Notes in Computer Science., Springer (2000) 162–177
4. MacKenzie, P.D., Shrimpton, T., Jakobsson, M.: Threshold password-authenticated key exchange. In Yung, M., ed.: Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings. Volume 2442 of Lecture Notes in Computer Science., Springer (2002) 385–400
5. Furukawa, J., Sako, K.: An efficient scheme for proving a shuffle. In Kilian, J., ed.: Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings. Volume 2139 of Lecture Notes in Computer Science., Springer (2001) 368–387
6. Neff, C.A.: A verifiable secret shuffle and its application to e-voting. In Reiter, M.K., Samarati, P., eds.: CCS 2001, Proceedings of the 8th ACM Conference on Computer and Communications Security, Philadelphia, Pennsylvania, USA, November 6-8, 2001, ACM (2001) 116–125
7. Weber, S.G., Araújo, R., Buchmann, J.A.: On coercion-resistant electronic elections with linear work. In: Proceedings of the The Second International Conference on Availability, Reliability and Security, ARES 2007, The International Dependability Conference - Bridging Theory and Practice, April 10-13 2007, Vienna, Austria, IEEE Computer Society (2007) 908–916
8. Clarkson, M.R., Chong, S., Myers, A.C.: Civitas: Toward a secure voting system. In: 2008 IEEE Symposium on Security and Privacy (S&P 2008), 18-21 May 2008, Oakland, California, USA, IEEE Computer Society (2008) 354–368
9. Araújo, R., Rajeb, N.B., Robbana, R., Traoré, J., Yousfi, S.: Towards practical and secure coercion-resistant electronic elections. In Heng, S., Wright, R.N., Goi, B., eds.: Cryptology and Network Security - 9th International Conference, CANS 2010, Kuala Lumpur, Malaysia, December 12-14, 2010. Proceedings. Volume 6467 of Lecture Notes in Computer Science., Springer (2010) 278–297
10. Spycher, O., Koenig, R.E., Haenni, R., Schläpfer, M.: A new approach towards coercion-resistant remote e-voting in linear time. In Danezis, G., ed.: Financial Cryptography and Data Security - 15th International Conference, FC 2011, Gros Islet, St. Lucia, February 28 - March 4, 2011, Revised Selected Papers. Volume 7035 of Lecture Notes in Computer Science., Springer (2011) 182–189
11. Rønne, P.B., Atashpendar, A., Gjøsteen, K., Ryan, P.Y.A.: Coercion-resistant voting in linear time via fully homomorphic encryption: Towards a quantum-safe scheme. CoRR **abs/1901.02560** (2019)
12. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In Franklin, M.K., ed.: Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings. Volume 3152 of Lecture Notes in Computer Science., Springer (2004) 41–55
13. Schweisgut, J.: Coercion-resistant electronic elections with observer. In Krimmer, R., ed.: Electronic Voting 2006: 2nd International Workshop, Co-organized by Council of Europe, ESF TED, IFIP WG 8.6 and E-Voting.CC, August, 2nd - 4th, 2006 in Castle Hofen, Bregenz, Austria. Volume P-86 of LNI., GI (2006) 171–177
14. Cachin, C., Kursawe, K., Shoup, V.: Random oracles in constantinople: Practical asynchronous byzantine agreement using cryptography. J. Cryptology **18**(3) (2005) 219–246
15. Furukawa, J., Sako, K.: An efficient publicly verifiable mix-net for long inputs. IEICE Transactions **90-A**(1) (2007) 113–127
16. Okamoto, T.: Provably secure and practical identification schemes and corresponding signature schemes. [36] 31–53

17. Chaum, D., Pedersen, T.P.: Wallet databases with observers. [36] 89–105
18. Yu, B., Liu, J.K., Sakzad, A., Nepal, S., Steinfeld, R., Rimba, P., Au, M.H.: Platform-independent secure blockchain-based voting system. In Chen, L., Manulis, M., Schneider, S., eds.: Information Security - 21st International Conference, ISC 2018, Guildford, UK, September 9-12, 2018, Proceedings. Volume 11060 of Lecture Notes in Computer Science., Springer (2018) 369–386
19. Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., Caro, A.D., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., Vukolic, M., Cocco, S.W., Yellick, J.: Hyperledger fabric: a distributed operating system for permissioned blockchains. In Oliveira, R., Felber, P., Hu, Y.C., eds.: Proceedings of the Thirteenth EuroSys Conference, EuroSys 2018, Porto, Portugal, April 23-26, 2018, ACM (2018) 30:1–30:15
20. Dimtiriou, T.: Efficient, coercion-free and universally verifiable blockchain-based voting. IACR Cryptology ePrint Archive **2019** (2019) 1406
21. Brickell, E.F., Camenisch, J., Chen, L.: Direct anonymous attestation. In Atluri, V., Pfitzmann, B., McDaniel, P.D., eds.: Proceedings of the 11th ACM Conference on Computer and Communications Security, CCS 2004, Washington, DC, USA, October 25-29, 2004, ACM (2004) 132–145
22. Gennaro, R., Gentry, C., Parno, B., Raykova, M.: Quadratic span programs and succinct nizks without pcps. In Johansson, T., Nguyen, P.Q., eds.: Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings. Volume 7881 of Lecture Notes in Computer Science., Springer (2013) 626–645
23. Boneh, D.: The decision diffie-hellman problem. In Buhler, J., ed.: Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings. Volume 1423 of Lecture Notes in Computer Science., Springer (1998) 48–63
24. Desmedt, Y., ed.: Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings. Volume 839 of Lecture Notes in Computer Science., Springer (1994)
25. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM J. Comput. **18**(1) (1989) 186–208
26. Blum, M., Feldman, P., Micali, S.: Non-interactive zero-knowledge and its applications (extended abstract). In Simon, J., ed.: Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA, ACM (1988) 103–112
27. Blum, M., Santis, A.D., Micali, S., Persiano, G.: Noninteractive zero-knowledge. SIAM J. Comput. **20**(6) (1991) 1084–1118
28. Boneh, D., Boyen, X.: Short signatures without random oracles. In Cachin, C., Camenisch, J., eds.: Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings. Volume 3027 of Lecture Notes in Computer Science., Springer (2004) 56–73
29. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret: Theory and applications of ring signatures. In Goldreich, O., Rosenberg, A.L., Selman, A.L., eds.: Theoretical Computer Science, Essays in Memory of Shimon Even. Volume 3895 of Lecture Notes in Computer Science., Springer (2006) 164–186
30. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Trans. Information Theory **22**(6) (1976) 644–654
31. Noether, S., Mackenzie, A.: Ring confidential transactions. Ledger **1** (2016) 1–18
32. Blanchet, B.: Automatic verification of security protocols in the symbolic model: The verifier proverif. In Aldini, A., López, J., Martinelli, F., eds.: Foundations of Security Analysis and Design VII - FOSAD 2012/2013 Tutorial Lectures. Volume 8604 of Lecture Notes in Computer Science., Springer (2013) 54–87
33. Delaune, S., Kremer, S., Ryan, M.: Verifying privacy-type properties of electronic voting protocols. Journal of Computer Security **17**(4) (2009) 435–487
34. Abadi, M., Blanchet, B., Fournet, C.: The applied pi calculus: Mobile values, new names, and secure communication. J. ACM **65**(1) (2018) 1:1–1:41

35. Dreier, J., Lafourcade, P., Lakhnech, Y.: A formal taxonomy of privacy in voting protocols. In: Proceedings of IEEE International Conference on Communications, ICC 2012, Ottawa, ON, Canada, June 10-15, 2012, IEEE (2012) 6710–6715
36. Brickell, E.F., ed.: Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings. Volume 740 of Lecture Notes in Computer Science., Springer (1993)