An analysis of a scheme proposed for electronic voting systems

Nicu NECULACHE [*] Vlad-Andrei PETCU [†] Emil SIMION [‡]

January 2023

Abstract

Voting mechanisms allow the expression of the elections by a democratic approach. Any voting scheme must ensure, preferably in an efficient way, a series of safety measures such as confidentiality, integrity and anonymity. Since the 1980s, the concept of electronic voting became more and more of interest, being an advantageous or even necessary alternative for the organization of secure elections. In this paper, we give an overview for the e-voting mechanisms together with the security features they must fulfill. Then we focus on the blind signature paradigm, specifically on the Pairing Free Identity-Based Blind Signature Scheme with Message Recovery (PF-IDBS-MR) [1]. Our goal is to give a better understanding on the PF-IDBS-MR scheme by offering an adaptation on the standard voting protocol's phases. More important, we analyze if the general security requirements and the recommendations proposed by the Council of Europe are met by the scheme.

Keywords: electronic voting, blind signature, PF-IDBS-MR

1 Introduction

The voting system is one of the key components of any democratic society. It allows people to freely express their elections anonymously. When organizing a voting session that fulfills all the security requirements, as privacy, anonymity, a lot of costs might be involved - time, materials, space etc. More than that, these sessions can be limited by geographical locations, or they can be corrupted by human actions.

With the evolution of technology, the implementation of electronic voting systems became of interest both for the academic environment and for most of the governments and democratic organizations. In November 2022, according to the International Institute for Democracy and Electoral Assistance [2], 49 countries already adopted different types of electronic systems for elections (the detailed statistics can be viewed in Figure 1). A technical approach solves most of the traditional system's issues, offering more accurate and faster election results. However, electronic voting seems to still face some challenges, especially due to security weaknesses, servers malfunction, and others.

First researches were started around 1980s (Chaum) [3]. Since then, numerous ideas and schemes have been proposed, each of them trying to cover the desired system needs. Consequently, it has arose

[*] Faculty of Computer Science, Alexandru Ioan Cuza University of Iasi, Email: nicu.neculache@gmail.com

[†] Faculty of Computer Science, Alexandru Ioan Cuza University of Iasi, Email: petcuvlad92@gmail.com

[‡] Politehnica University of Bucharest, Email: emil.simion@upb.ro

also the need to have a set of standards to be met by such a system. Council of Europe, started in 2004 to setup intergovernmental standards in the field of e-voting, resulting the Recommendation Rec(2004)11[4]. After a few years, in 2014 it was created and given the mandate to revise the standards. The new recommendation, which consists of the actual Recommendation CM/Rec(2017)5 [5] includes core aspects and guidelines on the implementation of e-voting systems. In this paper we highlight some of this principles in accordance with the scheme that we present.

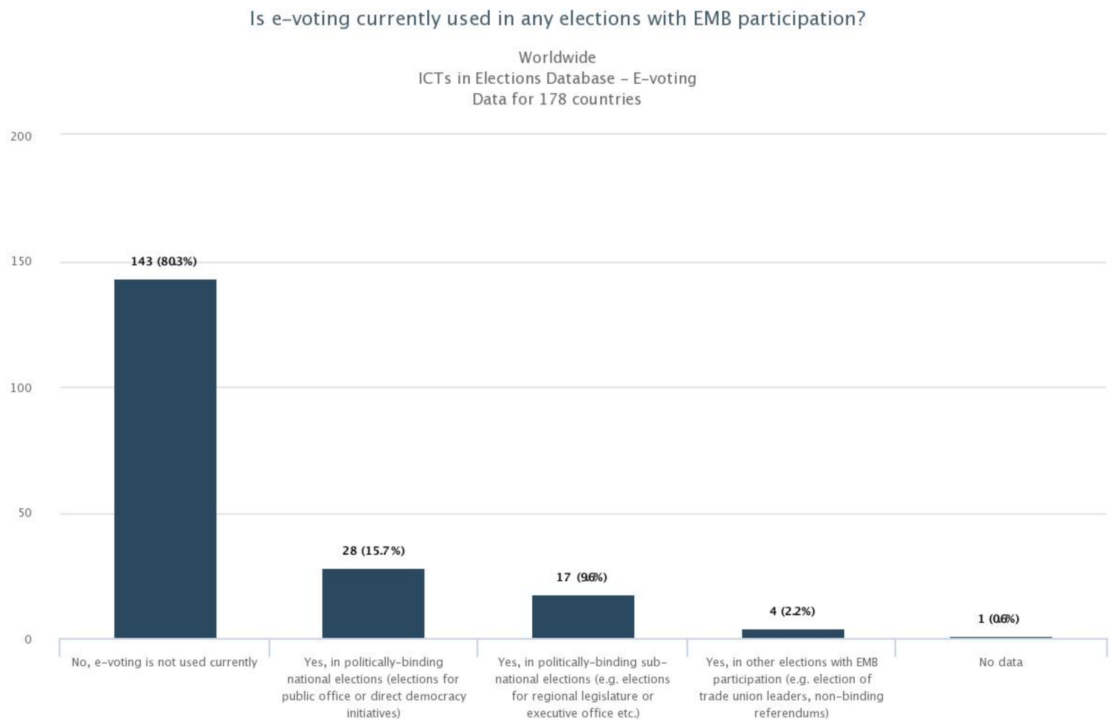

Figure 1: Worldwide E-voting Statistics [2]

2 Context

In this section we describe shortly how the voting system works. We present the phases of the voting protocol and the criteria that must be met by it.

2.1 The voting protocol

The voting protocol is based on a standard process composed of tree main phases:

1. **Setup**: The authorities prepare the voting system - includes nomination of candidates, computation of the list of candidates, registration of voters, and computation of the list of eligible voters.

2. **Voting**: The eligible voters exercise their right to vote - they authenticate, complete a ballot and finally submit it.

3. **Counting**: The authorities count the ballots and conclude the results.

2.2 General requirements

Any voting implementation must respect the above phases and satisfy the next essential properties:

1. **Eligibility**: Unqualified voters are not allowed to vote.
2. **Robustness**: Elections can not be disturbed by any malicious entity.
3. **Vote-privacy**: Ballots and the voting process events must remain secret.
4. **Integrity of the votes**: The election results cannot be falsified. Any voter can verify whether their vote is included in the final phase.
5. **Soundness, completeness and correctness**: The final tally included all valid ballots.
6. **Fairness**: To avoid any interference in voter behavior, counting cannot begin until the election is over.
7. **Dispute-freeness**: Any party can publicly verify whether the participant follows the protocol at any phase of the election.
8. **Transparency**: Maximise transparency in the vote casting, vote storing and vote counting process while preserving the secrecy of the ballots.
9. **Accuracy**: The system is errorless and valid votes must be correctly recorded and counted. These properties can be retained by universal verifiability.
10. **Accountability**: If the vote verification process fails, the voter can prove that he/she has voted, preserving at the same time the vote secrecy.

2.3 E-voting paradigms

Depending on the methods they use, e-voting protocols can be classified as homomorphic, mix-type or blind-signature based.

2.3.1 Homomorphic paradigm

This paradigm is making use of a homomorphic public key cryptosystem. A function $f : A \to B$ is homomorphic if:

$$f(x) \cdot f(y) = f(x + y)$$

Schemes in this paradigm either use additive or multiplicative homomorphisms. Most of the proposals use an additive homomorphic because it provides an efficient counting phase which requires only one decryption. Unfortunately, most of additive homomorphic cryptosystems use the factorization problem as a trapdoor, so their distributed key generation algorithms have an elevated cost.

2.3.2 Mix-type paradigm

Mix-type paradigm offers privacy by breaking the link between the ballots and the voters who cast them through a mixing process whose correctness has to be proven in zero-knowledge. The mixing phase implies a set of authorities, which are organized in a sequential manner. First one takes as input the ballots published on the bulletin board and mixes them (by permuting and re-encrypting). The second mixer then takes as input the output of the first authority and performs the same operation. This chain process is repeated for each mixer. In the end, a set of mixed ballots is obtained. Each mixing authority publishes its result together with a zero-knowledge proof. In this way, any entity can check its correctness. The most complex and time-consuming part of this paradigm is the generation and validation of the zero-knowledge proof of correct mixing.

2.3.3 Blind-signature paradigm

A blind signature protocol involves two parties, a message owner (Bob) and a signer (Alice). The message owner obtains a digital signature computed by the signer over his message while the signer obtains no information about the message that has to be signed.
The main steps of the blind signing process are the following:

1. Alice generates (sk, pk) - a private/public key pair
2. A random parameter r is used to disguise Bob's message m, resulting m' (i.e. a hash function can be used)
3. The hash m' is sent to Alice for signing (using sk), resulting s' (i.e. RSA can be used)
4. The digital signature s' is unblinded by Bob using r, resulting s (same as signing m with sk)

Given the message m, the signature s and the public key pk, it can be checked if s is a correct signature of m under the public key pk: $Ver_{pk}(m, s) \overset{?}{=} T/F$.

The usage of bling signatures ensures vote expression anonymously because it is not associated to the voter's signature. More than that, each ballot is signed by a Trusted Authority (TA), i.e. an Authentication Server, which ensures eligibility. When contacted by a voter, the TA checks that the voter appears in the electoral roll and then computes a blind signature of his ballot. That blindly signed ballot can then be submitted by the voter through an anonymous channel. In this way, the TA guarantees that all the ballots received by the collector, i.e. a Ballot Collection Authority, are cast by eligible voters.

3 PF-IDBS-MR Scheme

In this section we present the Pairing Free Identity based Blind Signature scheme proposed by Salome James, N.B. Gayathri and P. Vasudeva Reddy in [1]. The scheme is based on blind signatures and elliptic curves. Blind signatures provide anonymity and untraceability, while Elliptic Curve Cryptography (ECC)[6] ensures high security with smaller keys in size. More than that, because bilinear pairings and map to point hash functions require expensive operations, the scheme comes with message recovery in a pairing free environment. Message recovery is a concept where the message is embedded in the signature itself, either partial or full.

The system includes 4 phases: System Setup, Key Extraction, Blind Signature Generation and Blind Signature Verification, which can be associated with the 3 phases of the standard voting protocol described in Section 2.1. In Table 1. we listed the notations used.

Notation	Meaning
$x\|\|y$	The concatenation of x and y
$\oplus$	The x-or binary operation
$[x]_y$	Representation of x in base y
$\|x\|$	The number of bits needed to represent x
$\|x\|_y$	The first y bits of x from the right side
$_y\|x\|$	The first y bits of x from the left side

Table 1: Notation Meanings

3.1 Setup

The **system setup** consists of generating the parameters *params* and a master-key s. *params* are made public and s is kept secret. *params* are implicit input to all the following algorithms. For a given security parameter $k \in \mathbb{Z}^+$, a Public Key Generator (PKG) runs this algorithm as follows:

1. Choose a cyclic additive group G of prime order q with the points on an elliptic curve E and P as the generator of G.

2. Select $s \in Z_q^*$ randomly and compute the system public key $P_{pub} = s \cdot P$.

3. Choose the hash functions $H_1 : \{0,1\}^* \to Z_q^*$, $H_2 : \{0,1\}^* \to Z_q^*$, $H_3 : G \to \{0,1\}^{|q|}$ and $F1 : \{0,1\}^{l_1} \to \{0,1\}^{l_2}$, $F_2 : \{0,1\}^{l_2} \to \{0,1\}^{l_1}$, where l_1 and l_2 are positive integers such that $|q| = l_1 + l_2$.

4. The PKG publishes the system parameters $params = \{E, G, q, P, P_{pub}, H_1, H_2, H_3, F_1, F_2, l_1, l_2\}$.

3.2 Voting

In order to vote, any user must perform both following steps, key extraction and (blind) signature generation.

Key extraction:

1. The user sends its identity ID to the PKG (the user can be a voter or a signer).

2. When receiving the user's identity ID, the PKG chooses randomly $r \in Z_q^*$ and computes $R = r \cdot P$, $h_1 = H_1(ID, R, P_{pub})$, $d = (r + s \cdot h_1) \mod q$.

3. Finally, the ID's corresponding private key $D = (d, R)$ is sent to the user. This can be done via a secure channel.

Blind signature generation:

1. The voter wants to get $m \in \{0,1\}^{l_1}$ blindly signed by a signer, whose identity is ID_{signer}.

2. The signer chooses randomly $k \in Z_q^*$, computes $X = k \cdot P$ and sends (X, R) to the voter as a commitment.

3. The voter blinds the message m using the blinding factors $a, b \in Z_q^*$ randomly. After choosing the factors, the voter computes $\beta = F_1(m) \parallel (F_2(F_1(m)) \oplus m)$, $Y = a \cdot X + b \cdot \beta \cdot P$, $h_2 = H_2(ID_{signer}, R, Y)$, $\widetilde{h} = a^{-1} \cdot h_2 \mod q$ and sends $\widetilde{h}$ to the signer.

4. The signer uses his/her private key d to compute the signature $z_1 = (k + \widetilde{h} \cdot d)$, which is then sent to the voter.

5. In order to unblind the message, the voter computes $z_2 = (a \cdot z_1 + b \cdot \beta) \mod q$, $\alpha = H_3(ID_{signer}, z_2 \cdot P)$, $v = [\alpha \oplus \beta]_{10}$, obtaining (m, Y, R, v). $\omega = (Y, R, v)$ is the blind signature of the original message m.

3.3 Counting

The counting phase implies the **blind signature verification**, performed by the collector.

1. The voter sends its signature (vote) to the collector.

2. To verify the signature $\omega = (Y, R, v)$ for the message m and the identity ID_{signer}, the collector computes $h_1 = H_1(ID_{signer}, R, P_{pub})$, $h_2 = H_2(ID_{signer}, R, Y)$, $\widetilde{\alpha} = H_3(ID_{signer}, Y + h_2(R + h_1 \cdot P_{pub}))$, $\widetilde{\beta} = [v]_2 \oplus \widetilde{\alpha}$.

3. The possible recovered message $\widetilde{m}$ is computed as $\widetilde{m} = |\widetilde{\beta}|_{l_1} \oplus F_2({}_{l_2}|\widetilde{\beta}|)$.

4. If ${}_{l_2}|\widetilde{\beta}| = F_1(\widetilde{m})$, then the collector accepts the signature as valid and counts the vote $\widetilde{m}$ $(= m)$.

4 Analysis of the PF-IDBS-MR Scheme

In this section we analyze if the schema presented in Section 3 fulfils the general voting security requirements presented in Section 2.2. We also highlight some of the most recent recommendations proposed by the Council of Europe.

4.1 General voting security requirements

The signer is a Trusted Authority which holds the list of eligible voters. In this way, any voter can be verified when trying to get a signature. This ensures eligibility. More than that, the signer can discard from the beginning any attempt to fraud the elections. If the signer does not provide a valid signature for a possible malicious entity, then it can not disturb the elections. This ensures robustness.

Salome James et al. already demonstrated in their paper [1] Section 5 that the scheme ensures correctness, blindness and unforgeability. These properties guarantee vote-privacy, integrity, soundness, completeness and correctness for the voting system. The usage of blind signatures breaks the link between the voter's identity and the vote itself. As the signer signs blindly the vote, then the collector counts the valid signed votes, there is no need for the voter to submit its identity to the collector.

The correctness of the scheme ensures also accountability and accuracy. If the vote verification process fails due to any system error, the voter can prove that he/she has voted. He/she can resubmit the signature provided by the trusted signer in order to prove the vote validity. Given the fact that a voter can obtain a single signature and every ballot corresponds to a unique signature, any voter can resubmit his/her ballot, avoiding at the same time double-voting. So the system is errorless and valid votes can be correctly recorded and counted by the collector.

The rest of the properties can also be satisfied, anyway these are independent of the scheme, being dependent on the implementation of the whole system. Considering the security provided by the hardness of the elliptic curve discrete logarithm problem and the fact that no identity is linked to any signature, we can assume that every signature of the ballots can be made public. In this way transparency can be maximized and any voter can verify whether their vote is included in the final phase. Also, to avoid any interference in voter behavior, the collector can make the results public just when the election is over.

4.2 Recommendations proposed by the Council of Europe

The Recommendation CM/Rec(2017)5 Apendix I [5] contains a set of standards on e-voting which express objectives that e-voting must fulfil to conform to the principles of democratic elections and referendums.

Because the list of eligible voters is hold at the system's Trusted Authority, the system can uniquely identify the voters. The eligibility and robustness of the system, mentioned above, ensure *Standard No. 7 "Unique identification of voters..."*, *Standard No. 8 "The e-voting system shall only grant a user access..."* and *Standard No. 9 "The e-voting system shall ensure that only the appropriate number of votes..."* . Given that, only eligible voters can vote, they can vote once, and only their votes are included in the election result.

Standard No. 11 "It shall be ensured that the e-voting system presents an authentic ballot..." and *Standard No. 17 " The e-voting system shall provide sound evidence that each authentic vote..."* are guaranteed by the usage of blind signatures. Any voter needs to get its vote blindly signed by the signer, so the collector can properly recover and count it.

Standard No. 15 "The voter shall be able to verify that..." and *Standard No. 18 "The system shall provide sound evidence that only eligible voters'..."* are met due to the usage of elliptic curves in combination with the blind signatures. As mentioned above, we can assume that every signature of the ballots can be made public, so every voter can verify whether their vote is included in the final phase and whether only eligible voters have submitted ballots.

Standard No. 26 "The e-voting process, in particular the counting stage..." provides that it must not be possible to link the vote to the voter who cast it and thus prevents vote secrecy breaching. We consider this segregation tightly bound to the whole system design and it is achieved by separating the system authorities. The signer sign any vote blindly and sends the result back to the voter. The voter then unblinds the result and submits the vote with its signature. When receiving the vote, the collector checks its authenticity using the signer's identity, but no voter identity is provided at this stage.

The other standards, most of them related to accessibility, user experience, system response, etc., can be also met, but they remain the responsibility of the whole system implementation rather that the presented scheme.

5 Conclusions

The adoption of electronic voting systems comes with a major impact, as this seems to fix a lot of limitations of the traditional voting process. In the last years, more countries have already gain trust in choosing the electronic method in favour of the traditional one. As the requirements are outlined better and better, many schemes were proposed, everyone of them contributing as much on this evolution.

In this paper we presented the e-voting mechanism together with the main security features it must fulfil. We also presented the e-voting paradigms. The most important contribution of our work is represented by the PF-IDBS-MR scheme's adaptation on the standard voting protocol's phases, together with its analyses on the general security requirements and the recommendations proposed by the Council of Europe.

Following our analysis, we conclude that the PF-IDBS-MR scheme meets the criteria presented above and it seems to be a promising candidate for a standardized electronic voting system. Given that it respects the Recommendation CM/Rec(2017)5 [5] we believe that many countries could adopt the new standardized system more confidently. In this way, the electronic voting could became the main procedure of holding democratic elections and referendums.

In the future work, it would be interesting to demonstrate all the analysis using a real example. There are many programming languages nowadays that could be feasible for the system implementation. For example, *SageMath* [7] could be a good choice for the mathematical computation, especially for ECC. The implementation would be a more decisive step on the large scale adoption of the proposed system, so on the electronic voting.

References

[1] S. James, N.B. Gayathri and P. Vasudeva Reddy, *Pairing Free Identity-Based Blind Signature Scheme with Message Recovery*, https://doi.org/10.3390/cryptography2040029

[2] International Institute for Democracy and Electoral Assistence, November 2022, https://www.idea.int/data-tools/question-view/742

[3] Chaum, D.L., *"Untraceable Electronic Mail, Return Addresses, And Digital Pseudonyms. Commun"*, ACM 1981, 24, 84–90

[4] Council of Europe, *"Recommendation Rec(2004)11"*, September 2004, https://www.coe.int/t/dgap/goodgovernance/Activities/Key-Texts/Recommendations/Rec(2004)11_Eng_Evoting_and_Expl_Memo_en.pdf

[5] Council of Europe, *"Recommendation Rec(2017)5"*, June 2017, https://rm.coe.int/168071bc84

[6] Elliptic Curve Cryptography, https://csrc.nist.gov/Projects/elliptic-curve-cryptography

[7] SageMath - open-source mathematics software system, https://www.sagemath.org/

[8] A. C. Atanasiu, *"Securitatea Informatiei, Vol. 2 (Protocoale de Securitate)"*, INFODATA Cluj, October 2016