E-voting protocols in context of COVID19

Sfîrnaciuc Emilia* Vasilescu Miruna-Elena† Simion Emil‡

Abstract

Electronic voting consists of the methods that use an electronic system in the process of recording, counting or transmitting votes. It is relatively a new concept used in the democratic processes and especially in the context of COVID19. It's aim is to reduce errors and to improve the integrity of the election process. In this paper, we provide a review of the existing systems used in Europe. Initially, we mention the factors that influence the adoption of such systems at a large scale. We further describe the systems used in Russia (Moscow's primary) and in Romania (for counting the ballots). These systems are analyzed in order to find out if they respect technical challenges such as verifiability, dependability, security, anonymity and trust.

Keywords: e-voting, encryption, cryptanalysis, Blockchain, El-Gamal cryptosystem, Mu-Varadharajan system

1 Introduction

One of the main topics in today's society and especially in the Computer Science field is the use of electronic voting systems. These systems play a decisive role in democratic organizations and represent a relatively new technology that helps voters register their ballots in elections, using computerized systems.

Electronic voting means the methods that use an electronic system connected to the Internet in the process of recording, counting or transmitting votes. It can be used both for elections for political functions, as well as for democratic decisions and public opinion polls.

Since the publication of the first cryptographic protocol with applicability in electronic elections (Chaum , 1981 and 1982), numerous solutions have appeared in the academic environment to deal with security issues in this field. These issues are greater than ever due to the COVID19 pandemic. It accelerated the development of the digital age and created the imbalance between the operational side and the security side.

*Faculty of Computer Science, Alexandru Ioan Cuza University of Iași; Email:emilia.sfirnaciuc@info.uaic.ro

†Faculty of Computer Science, Alexandru Ioan Cuza University of Iași; Email:miruna.vasilescu@info.uaic.ro

‡Politehnica University of Bucharest: emil.simion@upb.ro

2 Context

2.1 The characteristics of an electronic system of vote

Every voting system used at a large scale should satisfy following properties:

– **Eligibility**: just legitimate voters can vote and only once.

– **Fairness**: no early results can be obtained which could influence the remaining voters.

– **Vote-privacy**: Ballots and all events during the voting process should remain secret.

– **Receipt-freeness**: a voter does not gain any information (a receipt) which can be used to prove to a coercer that she voted in a certain way.

– **Coercion-resistance**: a voter cannot cooperate with a coercer to prove to him that she voted in a certain way.

– **Integrity of the votes** (both voter verification, "I can check that my vote was captured correctly" and public verification, "anyone can check that all recorded votes were counted correctly");

– **Correctness of counting**: The final tally should be an accurate count of the ballots that have been cast.

2.2 Types of voting protocols

Voting protocols can be independent or central authority-based.

2.2.1 Independent protocol of voting

The simplest voting protocols do not use any authority, relying only on voters.

They are the first protocols designed for voting, based on successive applications of encryption and / or digital signing of messages. Anonymity is obtained by applying permutations in various phases.One of the most known independent voting protocols is developed by Michael Merritt.

2.2.2 Central authority-based protocols

The central authority (CA) is introduced to reduce the volume of computation. It's purpose is to register the electors.

There are some steps done in every central-authority-based protocol:

Assumption: each person k has two keys ek (public) and dk (secret).

1. AC asks each voter whether or not he or she wants to participate in the election.
2. The list of registered voters is published.
3. Each voter receives an ID from the CA (through a protocol of partial disclosure of secrets)
4. Each voter sends the pair $(ID; e_i(ID; m))$ to CA.

5. CA publishes $e_i(ID; m)$ for all voters on the list.

6. Each voter i anonymously sends the pair $(ID; d_i)$ to CA.

7. The CA associates the messages by ID, decrypts them, verifies the authenticity of the votes and publishes the pairs $(ID; m)$ for all participants in the vote.

This system prevents both unauthorized voters from voting and authorized voters from voting twice. Voters cannot find out their real identity, because each ID is obtained through a protocol of partial disclosure of secrets, so the CA does not know the ID reached by each voter.

Improvements:

There are also some problems in this protocol.

The first is that a central authority may be a point of corruption over which there is no control:

- it can falsify votes on behalf of abstaining voters,
- it can lose valid votes (no voter can prove that they really cast a vote)

In addition, its implementation remains quite complex.

A first idea to improve the system was to introduce several central authorities, dependent on each other. For example, two authorities can be introduced:

- LA - legitimacy agencies, that deals with the legitimacy of voters
- TA - tabulation agency, that deals with the actual counting.

A valid vote must be validated by both agencies.

- The first recognizes the right of the voter to vote (without seeing the content of the vote), issuing a ballot.
- The second agency receives the vote together with the validation ballot.

A variant of such a protocol (assuming that the two agencies do not ally to falsify the vote) is:

1. Each voter (after proving his / her identity) requests from LA an authentication number.

2. LA randomly generates authentication numbers and distributes them.

3. LA sends to TA the list of all authentication numbers.

4. Each voter randomly chooses an ID (validation number) and sends to TA a triplet consisting of his authentication number, ID and vote.

5. TA verifies the authentication number and - if it is on the list - ticks it and publishes the vote together with the validation number.

3 Mu-Varadharajan Protocol

The protocol is based on El Gamal and RSA digital signature schemes.

The components of the Mu-Varadharajan protocol are:

- V - a non-empty finite set of voters;
- AS - a voter authentication server;
- VS - a finite set of voting servers;
- TCS - a ballot counting server;
- CA - a certificate of authenticity.

The protocol contains three stages:
1. Obtaining the ballot paper;
2. Voting (and collection of ballot papers);
3. Counting of ballots.

3.1 Initialization

We will denote by p a large prime number and by t the time stamp. Also, $\alpha \| \beta$ will represent concatenation of sequences a and b.

Before starting the voting protocol:

- Each participant V receives a pair of RSA keys: $(e_v; d_v)$ and an n_v module obtained by multiplying two large prime numbers. According to the RSA algorithm:

$$e_v \cdot d_v \equiv 1 \,(\mathrm{mod}\Phi\,(n_V))$$

- AS has an n_{AS} module and an RSA key pair $(e_{AS}; d_{AS})$. n_{AS} and e_{AS} are public;

- Any valid voter V has a long-term $Cert_V$ voter certificate issued by the CA.

It is signed by the CA's secret key, and its contents include:
- a serial number
- the identity of voter V
- CA identity
- public key eV and module nV
- period of validity
- a stamp of time

Stage I: Obtaining the ballot

1. V must prove that he is a valid voter. For this, the voter chooses a blind factor b and three random numbers $g, r, k_1 \in Z_p^*$. Based on them, the parameters are calculated as follow:

$$a = g^r (\mathrm{mod} p)$$
$$x_1 = g b^{e_{AS}} \,(\mathrm{mod} n_{AS}) \quad (1)$$
$$x_2 = g^{k_1} b^{e_{AS}} \,(\mathrm{mod} n_{AS})$$
$$x_3 = a b^{e_{AS}} \,(\mathrm{mod} n_{AS}) \quad (2)$$

And sends to AS the pair $\left(V, AS, \text{Cert}_V, (x_1 \|x_2\| x_3\|t)^{d_v} \ (\text{mod}n_v)\right)$.

2. AS first checks the validity of the certificate and validates the signature $(x_1 \|x_2\| x_3\|t)^{d_v} \ (\text{mod}n_v)$. Then AS chooses a random number k_2 and calculates:

$x_4 = (k_2\|t)^{e_V} \ (\text{mod}n_V)$

$x_5 = \left(x_1^{3k_2} x_2^2 x_3\right)^{d_{AS}} \ (\text{mod}n_{AS}) = (y_1 y_2 a)^{d_{AS}} \ b^{3(k_2+1)} \ (\text{mod}n_{AS})$

where $y_1 = g^{k_1+k_2}$, $y_2 = g^{k_1+2k_2}$. The message $(AS, V, x_4, (x_5\|t)^{e_V} \ (\text{mod}n_V))$ is sent to V.

The parameter k_2 is different for each voter, and AS stores it in its database k2 with the identity of $V(Cert_V)$.

3. By decrypting x_4, V obtains $_2$. So V can calculate y_1 and y_2, then determine

$$s = x_5 b^{-3(k_2+1)} = (y_1 y_2 a)^{d_{AS}} \ (\text{mod}n_{AS})$$

s is the RSA signature for product $y_1 y_2 a$.

For a vote m, V can generate the ElGamal signature (s_1, s_2) :

$$s_1 = (k_1 + k_2)^{-1} \ (ma - r)(\text{mod}p - 1) \tag{3}$$

$$s_2 = (k_1 + 2k_2)^{-1} \ (ma - r)(\text{mod}p - 1) \tag{4}$$

V's ballot is:

$$T = a\|g\|y_1 \ \|y_2\| \ s \ \|s_1\| \ s_2\|m$$

Stage II: Voting (and collecting ballots)

In this phase V can send his ballot through the network to a VS voting server. The main purpose of a VS is to guarantee the validity of the ballot paper. The protocol contains two steps:

- V sends T to VS;

- VS decrypts T and verifies the validity of a, y_1, y_2 using his signature s and public key e_{AS}. Then VS determines the correctness of the signature (s_1, s_2) for m, using the relations:

$$a y_1^{s_1} = g^{ma} \tag{5}$$

$$a y_2^{s_2} = g^{ma} (\text{mod}p) \tag{6}$$

If the result of this check is positive, then VS has the certainty that the bulletin T is valid. VS stores all ballots and finally sends the database over the network to the counting server TCS.

Stage III: Counting the ballots

All VS's send the bulletins to TCS. Its purpose is to count the votes and to track down those who voted multiple times.

Assume that V uses the parameters a, g, k_1, k_2 to sign another vote m' and sends a second ballot a second ballot $T = a\|g\|y_1\,\|y_2\|\,s\,\left\|s_1'\right\|\,s_2'\|m'$ to another VS.

To detect a double vote, VS checks the parameters of a, g, y_1, y_2 of all T bulletins to see if they repeat. If so, he solves the linear system:

$$k_1 + k_2 = \frac{ma' - ma}{s_1' s_1} (\mathrm{mod} p - 1) \tag{7}$$

$$k_1 + 2k_2 = \frac{ma' - ma}{s_2' - s_2} (\mathrm{mod} p - 1) \tag{8}$$

And finds k_2.Using AS's database, voter V is uniquely identified.

3.2 Weaknesses of the Mu-Varadharajan protocol

Suppose that V has obtained the valid ballot $T = a\|g\|y_1\,\|y_2\|\,s\,\|s_1\|\,s_2\|m$. Based on it, it can generate another valid T' bulletin as follows:

3.2.1 First attack

At the beginning $\vee$ calculates g', y_1', y_2', a' with relationships

$$\begin{aligned} g' &= q^{c_0} (\mathrm{mod} p) \\ y_1' &= (g')^{(k_1+k_2+c_1)c_0^{-1}} (\mathrm{mod} p) \\ y_2' &= (g')^{(k_1+2k_2+c_2)c_0^{-1}} (\mathrm{mod} p) \\ a' &= (g')^{(r+c_3)c_0^{-1}} (\mathrm{mod} p) \end{aligned}$$

where c_0, c_1, c_2, c_3 integer numbers which satisfy the conditions $c_1+c_2+c_3 = 0$, $c_1c_2c_3 \neq 0$.

V generates the ballot $T' = a'g'y_1'\left\|y_2'\right\|\,s\,\left\|s_1'\right\|\,s_2'\|m$ where $\left(s_1', s_2\right)$ is the signature for the vote m with the keys $\left(k_1 + k_2 + c_1\right) c_0^{-1}$ respectively $\left(k_1 + 2k_2 + c_2\right) c_0^{-1}$:

$$s_1' = \left(\left(k_1 + k_2 + c_1\right) c_0^{-1}\right)^{-1} \left(ma' - \left(r + c_3\right) c_0^{-1}\right) (\mathrm{mod} p - 1)$$

$$s_2' = \left(\left(k_1 + 2k_2 + c_2\right) c_0^{-1}\right)^{-1} \left(ma' - \left(r + c_3\right) c_0^{-1}\right) (\mathrm{mod} p - 1)$$

In the voting and ballot collection phase, V may send to VS and the second ballot T'. VS first check the signature s, then the validity of y_1', y_2', a' using the equation

$$s^{e_{AS}} = y_1 y_2 a\,(\mathrm{mod} n_{AS}) = y_1{}' y_2' a'\,(\mathrm{mod} n_{AS})$$

Next VS checks the validity of $(s_1{}', s_2)$ with equations (5) and (6). If all the checks pass, VS considers that T' is a valid bulletin and sends it to TCS. TCS checks the parameters g', y_1', y_2', a ' and validates that they have been used only once (protection against double voting).

So the attack works. Even if VS detects that the signature s has been used before, he will not be able to detect the identity of the illegal voter (with (7) , and (8)).

3.2.2 Second attack

Similarly to the first attack, V first randomly chooses the number h and calculates:

$$g' = g^h, y_1' = y_1^{h^2}, y_2' = y_2^{h^2}, a' = a^{h^2}, s' = s^{h^2}$$

Then the signature (s_1', s_2') for m can be calculated with a variant of relations (3) and (4) using the keys $(k_1 + 2k_2)\, h$ and $(k_1 + 2k_2)\, h$:

$$s_1' = (k_1 + k_2)^{-1} h^{-1} (ma' - hr) \,(\mathrm{mod} p - 1)$$
$$s_2' = (k_1 + 2k_2)^{-1} h^{-1} (ma' - hr) \,(\mathrm{mod} p - 1)$$

V is able to generate a new ballot $T' = a' \,\|g'\|\, y_1' \,\left\|y_2'\right\|\, s' \,\left\|s_1'\right\|\, s_2' \| m$. The following relations are satisfied:

$$(s')^{e_{AS}} = (y_1 y_2 a)^{h^2} = y_1' y_2' a' \,(\mathrm{mod} n_{AS})$$
$$\left(y_1'\right)^{s_1'} a' = (g')^{a'm} \,(\mathrm{mod} p)$$
$$\left(y_2'\right)^{s_2'} a' = (g')^{a'm} \,(\mathrm{mod} p)$$

Due to that, VS is certain about the validation of T '.

These attacks allow a voter to cast as many votes as he wants, without being detected.

3.3 Improvement over the Mu–Varadharajan e-voting protocol

In order to avoid weaknesses of the attacks presented above, an improved version of the Mu-Varadharajan voting protocol was developed. The new scheme also has three stages.

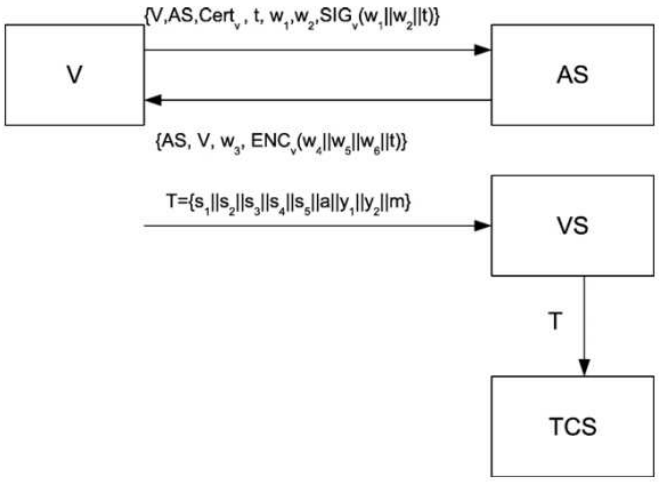

FIGURE 1: Scheme proposed by Lin-Hwang-Chang [16]

Stage I: Obtaining the ballot

A. V picks two blind factors b_1, b_2 and two random numbers k_1, r. With these parameters

V will calculate w_1 and w_2 as follows:

$$w_1 = g^r b_1^{e_{AS}} \,(\mathrm{mod} n_{AS})$$
$$w_2 = g^{k_1} b_2^{e_{AS}} \,(\mathrm{mod} n_{AS})$$

Where $g \in Z_p^*$ is a public parameter of the system.

V sends to AS : $\left\{V, AS, Cert_V, w_1, w_2 \left((w_1 \,\|w_2\|\, t)^{d_v} \,(\mathrm{mod} n_v)\right)\right\}$

B. VS checks the validity of the certificate and validates the signature $(w_1 \,\|w_2\|\, t)^{d_v} \,(\mathrm{mod} n_v)$

If it passes the check, AS is sure about the correctness of the received parameters.

It continues to pick a random number k_2 which is different for each voter and calculates:

$$w_3 = (k_2 \| t)^{e_V} \,(\mathrm{mod} n_V)$$
$$w_4 = (w_1 \cdot AS)^{d_{AS}} \,(\mathrm{mod} n_{AS}) = (a \cdot AS)^{d_{AS}} b_1 \,(\mathrm{mod} n_{AS})$$
$$w_5 = \left(w_2 \cdot g^{k_2} \cdot AS\right)^{d_{AS}} \,(\mathrm{mod} n_{AS}) = (y_1 \cdot AS)^{d_{AS}} \, b_2 \,(\mathrm{mod} n_{AS})$$
$$w_6 = \left(w_2^2 \cdot g^{k_2} \cdot AS\right)^{d_{AS}} \,(\mathrm{mod} n_{AS}) = (y_2 \cdot AS)^{d_{iS}} \, b_2^2 \,(\mathrm{mod} n_{AS})$$

Where $a = g^r, y_1 = g^{k_1+k_2}, y_2 = g^{k_1+2k_2}$.

The message $\left\{AS, V, w_3 \left((w_4 \,\|w_5\|\, w_6 \| t)^{d_v} \,(\mathrm{mod} n_v)\right)\right\}$ is sent to V. It also saves in the database k_2 with the identity of V.

C. V obtains k_2 decrypting w_3 and can determine y_1 and y_2. V computes the signature (s_1, s_2, s_3) as follows:

$$s_1 = w_4 b_1^{-1} = (a \cdot AS)^{d_{AS}} quad \,(\mathrm{mod} n_{AS})$$
$$s_2 = w_5 b_2^{-1} = (y_1 \cdot AS)^{d_{AS}} \, quad \,(\mathrm{mod} n_{AS})$$
$$s_3 = w_6 b_2^{-2} = (y_2 \cdot AS)^{d_{AS}} \, quad \,(\mathrm{mod} n_{AS})$$

D. V applies an ElGamal signature scheme to sign the vote m.

Given y_1, y_2 the public keys of the system and $x_1 = k_1 + k_2, x_2 = 2k_1 + k_2$ the corresponding secret keys, so $y_1 = g^{k_1+k_2} (\mathrm{mod} p)$ and $y_2 = g^{2k_1+k_2} (\mathrm{mod} p)$. The signature $((a, s_4), (a, s_5))$ of the vote m is generated:

$$s_4 = x_1^{-1}(ma - r) (\mathrm{mod} p - 1)$$
$$s_5 = x_2^{-1}(ma - r) (\mathrm{mod} p - 1)$$

The ballot for the voter V is $T = s_1 \,\|s_2\|\, s_3 \,\|s_4\|\, s_5 \| a \| y_1 \,\|y_2\|\, m$

Stage II: Voting (and collecting ballots)

A. V sends the ballot T to VS.

B. VS verifies the validation of a, y_1, y_2 using the following equations:

$$AS \cdot a = s_1^{e_{AS}} \,(\mathrm{mod} n_{AS}) \tag{9}$$

$$AS \cdot y_1 = s_2^{e_{AS}} \,(\mathrm{mod} n_{AS}) \tag{10}$$

$$AS \cdot y_2 = s_3^{e_{AS}} \,(\mathrm{mod} n_{AS}) \tag{11}$$

If all of them are checked, VS proceeds to check the correctness of the signature $((a, s_4), (a, s_5))$ using:

$$g^{ma} = y_1^{s_4} a = y_2^{s_5} a (\mathrm{mod} p)$$

If checked, VS accepts T as valid. Finally, VS forms a database with all of the valid ballots, which he sends through the TCS network.

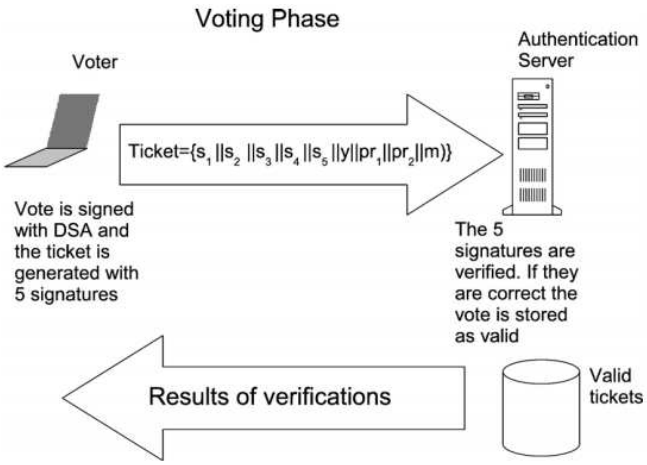

FIGURE 2: Second phase of the proposed scheme [8]

Stage III: Counting the ballots

TCS will publish and count all the votes, after receiving them from allVS. It is also responsible for the detection of double voting.

If a voter V uses the same parameters a, y_1, y_2 to sign a different vote m ' and sends this ballot to another VS :

1.TCS verifies a, y_1, y_2 for all the ballots T in order to check if they appear twice.

2. Only one ballot is taken into consideration if these values appeared twice and $m = m'$. If $m \neq m'$, it is double voting tentative and TCS compute his identity:

$$x_1 = \frac{ma' - ma}{s_4' - s_4} (\mathrm{mod} p - 1)$$
$$x_2 = \frac{ma' - ma}{s_5' - s_5} (\mathrm{mod} p - 1)$$

$$x_2 - x_1 = (2k_1 + k_2) - (k_1 + k_2) = k_1$$
$$k_2 = x_1 - k_1$$

The voter who tried to cheat is identified using k_2.

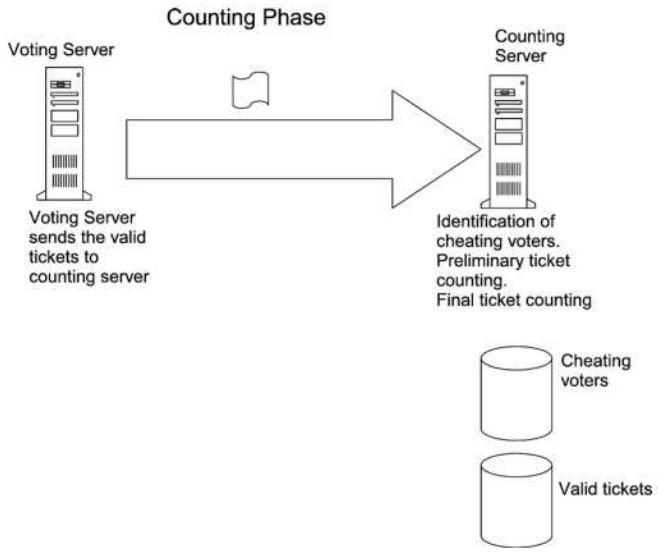

FIGURE 3: Third phase of the proposed scheme [8]

3.4 Security of the modified Mu-Varadharajan protocol

The built system verifies all the restrictions of an electronic voting system if the SA authentication server is secure and therefore will not generate any ballot without the consent of the voter. In addition, it has increased security as follows:

Resistance to attacks 1 and 2

Suppose a voter forces the parameters a, y_1, y_2 according to Attack 1; he will not be able to obtain the signature (s_1, s_2, s_3) given by equations (9); (10); (11) because he does not know the secret key d_{AS}.

In the second attack, $\vee$ can easily obtain $s_2' = \left(AS \cdot y_2'\right)^{d_{AS}}$ but cannot generate the rest of the signature for the vote m. For example, suppose $s_2{}' = s_2^2 = \left(AS^2 \cdot y_2^2\right)^{d_{AS}}$ and $y_2' = AS \cdot y_2^2$. So the parameter y_2 can pass the verification. But - due to the problem of discrete logarithms - the voter cannot obtain the secret key corresponding to $x_1{}'$. Without the secret key, V cannot generate a correct signature.

Resistance to an allied attack

Suppose two V_1, V_2 voters with the valid signatures (s_{11}, s_{12}, s_{13}) respectively (s_{21}, s_{22}, s_{23}) that collaborates to obtain a new signature $(s_1{}', s_2{}', s_3{}')$ defined by

$$s_i' = s_{1i} \cdot s_{2i} \,(\mathrm{mod} n_{AS}), \quad i = 1, 2, 3, 4$$

However, they will not be able to calculate the parameters r', x_1', x_2' due to the difficulty of solving the discrete logarithm problem.

4 Usage of e-voting

Some countries that partially use e-voting are France, Belgium, Russia, Bulgaria (as shown in figure 4).

Estonia is the first nation who successfully implemented e-voting for the municipal elections in 2005 [2]. From then, the number of individuals voting over the Internet has been increasing consistently. The voting system uses an ID card which represents a mandatory national identity document as well as a smart card. It ensures both secure remote authentication and legally binding digital signatures by using the public key infrastructure [6].

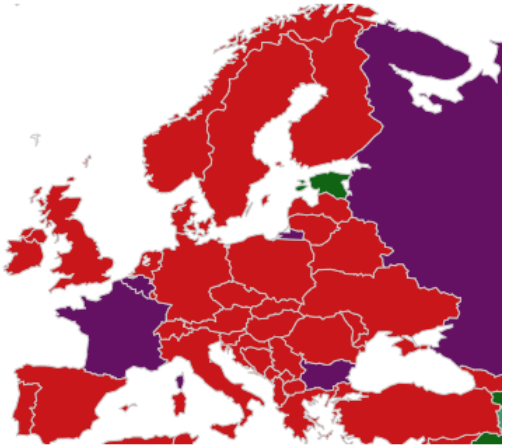

FIGURE 4: Usage of e-voting in Europe [1]

Internet voting is available prior to Election Day from 10th day to 4th day. Voters can change their electronic votes an unlimited number of times. It is also possible for anyone who votes using the Internet to vote at a polling station during the early voting period, invalidating their Internet vote. Only the final vote is counted.

Romania first implemented electronic voting systems in 2003 for the national referendum. The purpose was to extend voting capabilities to military forces serving in Afghanistan, Iraq and Kosovo (theaters of war). Despite the publicly stated goal of fighting corruption, the equipment was procured and deployed in less than 30 days after a government edict passed [1].

According to some changes in the electoral law (March 2020), Russia plans in the close future to use e-voting or correspondence voting for every type of election. One attempt was made in Moscow (2019) which was unsuccessful with the respect of security requirements. Some people believe that the system was the starting point for stopping the extension of COVID19 (no need of physical presence) and the opponents believe that it will be much easier to manipulate the system.

5 Case Study: Moscow's e-voting system

In 2019, Moscow's information technology department developed a remote voting system using blockchain. The encryption used in this system is a variant of ElGamal over finite fields. Before the election day, the system was updated due to some tests and attacks found after code publication (as shown in the figure 5).

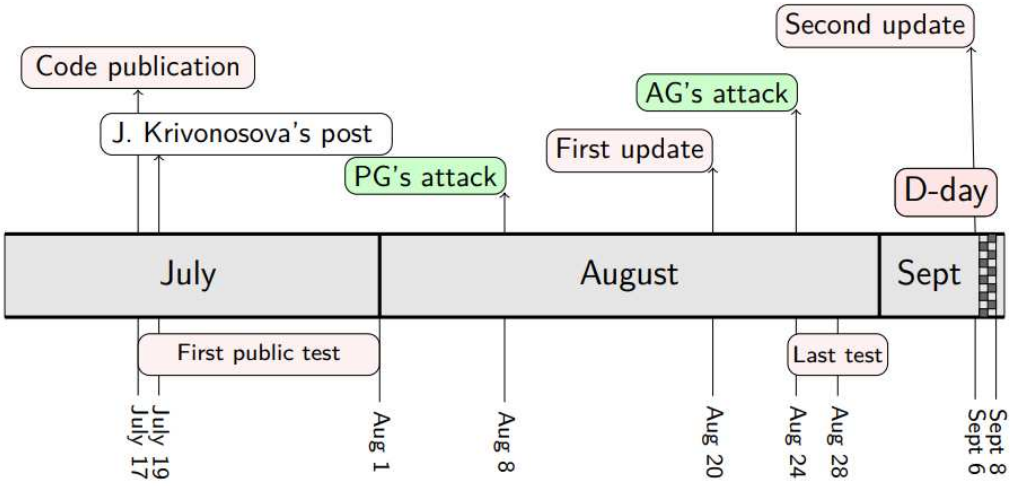

FIGURE 5: Evolution of Moscow's e-voting system [7]

5.1 Criptographic fundamentals

ElGamal cryptosystem

Key generation: Let g be a generator, and $(sk, pk = g^{sk})$ a key-pair.

Encryption: The plain ElGamal encryption of a message m is:

$$\text{Enc}_{g,pk}(m) = (a, b) = g^r, pk^r \cdot m$$
$$\text{Enc}_{g,pk}(m) = (a, b) = (g^r, \text{pk}^r \cdot m)$$

where r is a random (to be used only once).

Decryption: The decryption using sk is:

$$\text{Dec}_{g,\text{sk}}(a, b) = b \cdot a^{-\text{sk}} = m$$

A triple-ElGamal

Encryption:

Choose three safe primes: $p_1 < p_2 < p_3$, with 3 generators g_1, g_2, g_3 for each element.

The keys are:

$$\text{sk} = (\text{sk}_1, \text{sk}_2, \text{sk}_3); \quad \text{pk} = \left(g_1^{\text{sk}_1}, g_2^{\text{sk}_2}, g_3^{\text{sk}_3}\right)$$

The encryption of a message $m \in Z/p_1Z$ is obtained by

$$(a_1, b_1) := \text{Enc}_{g_1 pk_1}(m); \text{map } a_1 \text{ to } Z/p_2Z$$
$$(a_2, b_2) := \text{Enc}_{g_2 pk_2}(a_1); \text{map } a_2 \text{ to } Z/p_3Z$$
$$(a_3, b_2) := \text{Enc}_{g_3 pk_3}(a_2)$$

and the encrypted message is

$$\text{MultiEnc}\ (m) = (b_1, b_2, a_3, b_3)$$

Decryption: Knowing sk and the inequality $p_1 < p_2 < p_3$, the operations can be reversed to decrypt m from $(b1, b2, a3, b3)$

$$a_2 := \mathrm{Dec}_{g_3 sk_3}\,(a_3 b_3)\,;\mathrm{map}\,a_2 \text{ to } Z/p_2 Z$$
$$a_1 := \mathrm{Dec}_{g_2 sk_2}\,(a_2 b_2)\,;\mathrm{map}\,a_1 \text{ to } Z/p_1 Z$$
$$m := \mathrm{Dec}_{g_1 sk_1}\,(a_1 b_1)$$

Security

Due to time and complexity constraints the contributors decided to increase the security of the system by using the Triple El-Gamal technique instead of writing a new library in Solidity[1]. All the p_i's are chosen to be less than 256 bits because this is the max size of native integers [7].

5.2 Weaknesses in the Moscow Internet voting system

First attack:

A French security researcher found a critical vulnerability in the system: he could compute the voting system's private keys based on its public keys. They were too small to be secure. This meant that modern computers could break the encryption scheme within minutes.

An official agreed that 256x3 private key length is not secure enough and that it will be changed to 1024. However, a public key of a length of 1024 bits may not be enough, according to Gaudry, who believes officials should use one of at least 2048 bits instead [7].

The goal of the attack was to compute a discrete logs mod a p of 256 bits. After this remark, the first fix was done.

In this updated version made on August 20 they removed the triple-ElGamal encryption and they upgraded the key-size to 1024 bits.

Second attack:

The second attack was made by Golovnev [5]. He observed that the generator is now in a prime-order subgroup but the messages are not. From the source code it appears that the encrypted message is the real identifier of a candidate (no random nonce was used).

When they realised that the system is vulnerable to this kind of attack, they denied its existence. They silently changed the code updating GitHub only two days before the election.

Another issue in the system was the shared key between the voter and the election committee which can be used for encrypting the vote, as well as decrypting it. That means

[1]Solidity is the smart-contract language of Ethereum.

that a Moscow voter can not only send their encrypted vote to the election commission's server (where it will enter into the blockchain database), but also manage to decrypt it without having to wait for the election commission to start the official vote count.

Secret keys for encrypted votes being readily available to voters could put them at risk of coercion. The voter could also record the online voting process in a HAR file. That file type allows users to save all incoming and outgoing traffic as they visit a web site or use a web app. Within that archive, you can find the user's encrypted vote.

5.3 The election day

About 10,000 votes were registered by the Internet and in a district, the difference between the first and the second candidate was less than 100 votes [5].

Once voting had concluded and a final CSV file containing data about 9,810 votes had been published, Moscow officials reconstructed a private key to decrypt those votes. The key had to be pieced together from fragments that had been entrusted to seven different officials in advance [3]. Unexpectedly, the reconstructed key (needed to decrypt the data) was entered into the voting experiment's public blockchain (as shown in the figure 6).

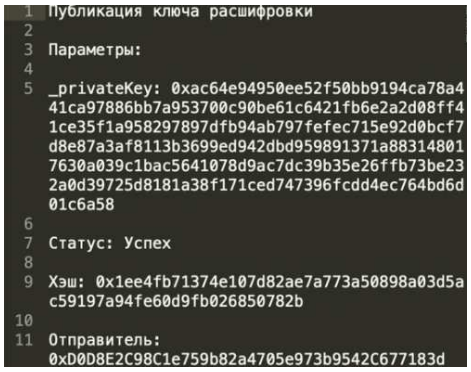

FIGURE 6: Example of reconstructed private key [3]

To decrypt the transactions in the blockchain, it needed more than the private key: one component of the public key that had been used to encrypt the data in the first place: its modulo. In order to find it, the HTML page for a ballot was used due to the fact that the page contained the public key and the modulo (as shown in the figure 7).

FIGURE 7: Part of HTML page for a ballot [3]

5.4 Conclusions

Due to these vulnerabilities, Moscow's information technology department will have to change its online voting encryption system completely once again [4]. Moscow's system does not satisfy the main security principles (privacy, coercion resistance, vote buying) and it is partially verifiable.

These were the first attacks made on the system but they may be multiple other ones if the system will be made entirely public in the future. It is possible that the voters used paper ballots during the election day because the system was hardly criticized by the press.

6 Voting trends in the context of COVID-19

Many experts have stated that limitations of current technology and internet infrastructure present unacceptable risks to elections integrity during the 2020 election cycle. Some say that risk inherent to internet technology and personal computing devices—specifically, the prevalence of malware—is a major barrier to its use in future elections, while others allow for the possibility that emerging technology and social acceptance of certain inherent risks may ultimately allow for its use in U.S. elections.

Academic researchers have published open letters to Congress and state officials to voice security concerns about online voting. Some states have cancelled or altered plans to expand internet voting pilot programs in response to critical third-party security assessments.

A case study has been done in the German state of Bavaria with an all-postal vote for the second round of local elections.

Bavaria held the first round of local elections on 16 March 2020, offering in person voting at polling stations and postal voting. The second-round run-off, held on 29 March 2020, was an all-postal vote. That was decided in response to the COVID-19 pandemic and the health risk it posed of contagion through social contact. The decision and logistical arrangements were made after the first round was held. [10]

This demonstrates that postal voting is preferred instead of e-voting due to security risks. In May 2020, CISA (Cybersecurity and Infrastructure Security Agency) and other federal agencies released guidance to the states advising them to limit use of electronic ballot return systems due to "significant security risks to the confidentiality, integrity, and availability of voted ballots." [9]

7 Blockchain technology used for parliamentary elections in Romania

The Special Telecommunications Service (STS) developed a blockchain technology system used in the parliamentary elections.This state-of-the-art complementary solution guarantees integrity and reinforces the transparency and traceability of the electoral process [11].

This technology does not allow the modification or alteration of the data recorded during the electoral process, not even by their administrators. The data from SIMPV[2] and SICPV[3]

[2]SIMPV – Information System for Monitoring Turnout and Preventing Illegal Voting
[3]SICPV – Information System for Centralizing Reports

can be verified by anyone on the dedicated website - https://voting.roaep.ro [12], where the registered information and their details are presented in real time. Both systems signal the attempts of illegal voting and prevent multiple voting.

Blockchain technology consists in calculating unique and unrepeatable digital fingerprints, which are updated every five seconds. A possible change in the information generates a new fingerprint, making the change visible. Due to that, blockchain technology forms a chain of trust in the flow of information.

7.1 Technical specifications regarding BlockChain based system

Below is an example of registered ballot: the previous hash is part of the current vote in order to obtain the hash chain. Other details are the time of creation, the root and the current hash, and also information about the signature.

The HASH function used is BLAKE2b[13] which is faster than MD5, SHA-1, SHA-2, and SHA-3, on 64-bit x86-64 and ARM architectures. BLAKE2 removes addition of constants to message words from BLAKE round function, changes two rotation constants, simplifies padding, adds parameter block that is XOR'ed with initialization vectors, and reduces the number of rounds from 16 to 12 for BLAKE2b. [14]

FIGURE 8: Registered ballot example [12]

Ed25519 is the EdDSA signature scheme using SHA-512 (SHA-2) and Curve25519. As security features, Ed25519 does not use branch operations and array indexing steps that depend on secret data, so as to defeat many side channel attacks. It is a public-key signature system with several attractive features: fast single-signature verification, very fast signing and key generation, high security level (it has a 2^{128} security target). Another big advantage is collision resilience: hash-function collisions do not break this system. [15]

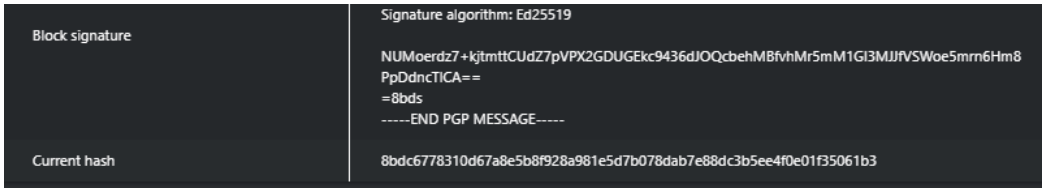

FIGURE 9: Block signature and current hash example [12]

7.2 Performance of BlockChain based system

The signature of the ballot can also be verified. The software takes only 273364 cycles to verify a signature on Intel's widely deployed Nehalem/Westmere lines of CPUs.

Nehalem and Westmere include all Core i7, i5, and i3 CPUs released between 2008 and 2010, and most Xeon CPUs released in the same period. The software performs a batch of 64 separate signature verifications (verifying 64 signatures of 64 messages under 64 public keys) in only 8.55 million cycles, under 134000 cycles per signature [12]. The software fits easily into L1 cache, so contention between cores is negligible: a quad-core 2.4GHz Westmere verifies 71000 signatures per second, while keeping the maximum verification latency below 4 milliseconds. For the parliamentary elections held in 2020 in Romania it takes one hour and 37 minutes to validate all the votes.

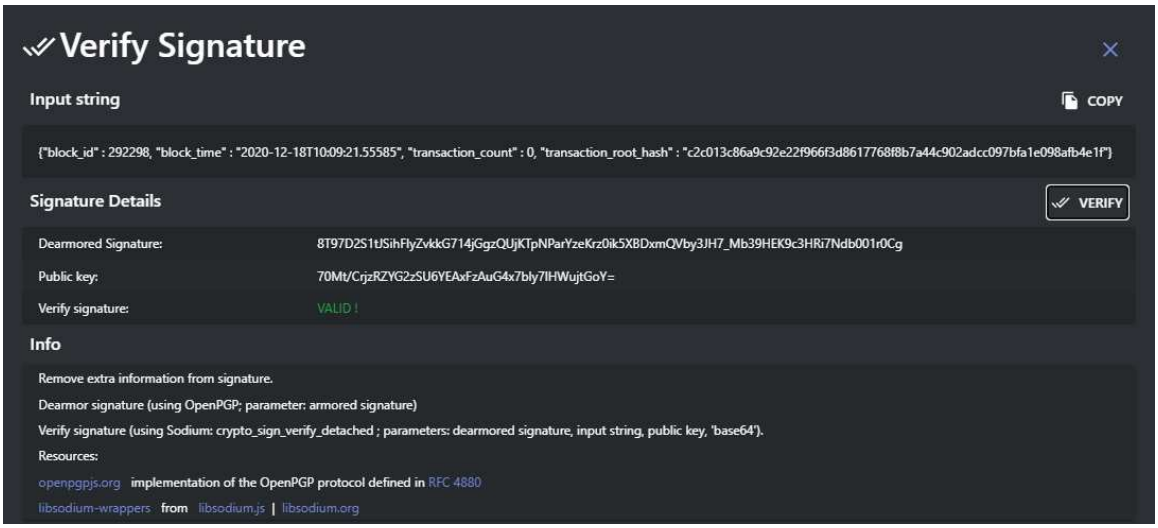

FIGURE 10: Signature verification example [12]

8 Conclusions

E-voting represents all the electronic means used for at least one of the following steps: recording, casting or counting of votes. There is no universal solution regarding the problem of security in e-voting. So, the voters are very skeptical as respects the secret of their ballot. It is unacceptable to make public the decrypted votes, which is the major

problem in Moscow's system. On the other hand, a good thing was the code made public, in order to have a better coverage on testing. As far as the counting of ballots goes, in the romanian e-voting system we were not able to find the source code and the only way to examine this application was through the encrypted votes displayed in GUI.

As this technique of voting becomes more popular, there are multiple existent solutions especially for counting phase. But the effective recording of voting is more complex and it requires a thorough analysis before introducing e-voting in society as the main way of voting.

References

[1] *Is E-Voting currently used in any election with EMB participation?* , https://www.idea.int/data-tools/question-view/742

[2] *Potential and challenges of E-voting in the European Union* (2016) , https://www.europarl.europa.eu/RegData/etudes/STUD/2016/556948/IPOL_STU(2016)556948_EN.pdf

[3] *Technical difficulties Possibly by accident, Moscow officials released the decryption key for the city's online votes* (2020) , https://meduza.io/en/slides/technical-difficulties?fbclid=IwAR1NbszSXM-dkV2tuoXtjeS-ijKskyC10NCizu3sCeWbHKHUuy3YwxMLnU4

[4] Meduza (2020) , *The real Russia today* https://meduza.io/en/feature/2020/07/02/moscow-s-online-voting-system-has-some-major-vulnerabilities-allowing-votes-to-be-decrypted-before-the-official-count

[5] Gaudry P., Alexander Golovnev A. (2019). *Breaking the encryption scheme of the Moscow Internet voting system* , https://arxiv.org/abs/1908.05127

[6] *Estonian Internet voting* (2019) https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/2019/07/29/Estonian+Internet+voting

[7] Gaudry P. (2020). *Weaknesses in the Moscow Internet voting system, RWC 2020* , https://rwc.iacr.org/2020/slides/Gaudry.pdf

[8] Rodríguez-Henríquez, F., Ortiz-Arroyo, D., & García-Zamora, C. (2007). *Yet another improvement over the Mu-Varadharajan e-voting protocol. Computer Standards & Interfaces* , https://sci-hub.do/10.1016/j.csi.2006.11.003

[9] Brian E. Humphreys (2020), *COVID-19: Remote Voting Trends and the Election Infrastructure Subsector*

[10] *International IDEA Technical Paper - Elections and COVID-19* (2020), https://www.idea.int/sites/default/files/publications/elections-and-covid-19.pd

[11] *Blockchain technology guarantees the integrity of IT systems for parliamentary elections* (2020) , https://www.sts.ro/ro/anunturi/tehnologia-blockchain-garanteaza-integritatea-sistemelor-informatice-pentru-alegerile-parlamentare

[12] *BlockChain - counting ballots official page*, https://voting.roaep.ro/

[13] *BLAKE2 — fast secure hashing*, https://www.blake2.net/

[14] Aumasson, Jean-Philippe; Neves, Samuel; Wilcox-O'Hearn, Zooko; Winnerlein, Christian (2013), *BLAKE2: simpler, smaller, fast as MD5*

[15] Daniel J. Bernstein (2017), *Ed25519: high-speed high-security signatures*

[16] I. Lin, M. Hwang, C. Chang (2003), *Security enhancement for anonymous secure e-voting over a network, Comput. Stand. Interfaces 25 (2) 131–139.*