# Russian Federal Remote E-voting Scheme of 2021 – Protocol Description and Analysis

Jelizaveta Vakarjuk[1,2], Nikita Snetkov[1,2,3], and Jan Willemson[1]

[1] Cybernetica AS, Mäealuse 2/1, 12618 Tallinn, Estonia,
{jelizaveta.vakarjuk,nikita.snetkov,jan.willemson}@cyber.ee,
[2] STACC OÜ, Narva mnt 20, 51009 Tartu, Estonia,
[3] Tallinn University of Technology, Akadeemia tee 15a, 12618 Tallinn, Estonia

**Abstract.** This paper presents the details of one of the two cryptographic remote e-voting protocols used in Russian parliamentary elections of 2021. As the official full version of the scheme has never been published by the election organisers, our paper aims at putting together as complete picture as possible from various incomplete sources. As all the currently available sources are in Russian, our presentation also aims at serving the international community by making the description available in English for further studies. In the second part of the paper we provide an initial analysis of the protocol, identifying the potential weaknesses under the assumptions of corruption of the relevant key components. As a result we conclude that the biggest problems of the system stem from weak voter authentication. In addition, as it was possible to vote from any device with a browser and Internet access, the attack surface was relatively large in general.

## 1 Introduction

Usage of electronic means to support the process of voting has been a subject of extensive research and debates. On one hand, computerised systems help keep better records of eligible voters, and election results will most probably find their way into some database.

Supporting the act of vote casting with electronic means is, however, a separate issue. For example, the history of Direct Recording Electronic voting equipment is rich in poor design choices and the resulting vulnerabilities [12, 4, 14, 1, 3, 2].

Voting over Internet has been even more controversial. In today's increasingly mobile world some sort of a remote voting option is necessary. Adding to it, the current COVID-19 pandemic has made large-scale high-contact events like election days disadvantageous. The classical alternative of paper-based postal voting has several drawbacks including weak voter authentication, potentially unreliable postal services, and the difficult-to-control threat of coercion. By using the appropriate technical and cryptographic means, all these issues can be addressed more easily in the case of remote electronic (Internet) voting.

On the other hand, Internet voting also brings along certain risks. The voter's perception of the voting environment is indirect (physical booth and paper *vs.* internals of a computer), leaving more room for e.g. malware to operate undetectably. On the server side, many operations are centralised, creating lucrative target points for attackers.

Whereas the debate over the manageability of these risks is still ongoing, several countries have experimented with remote electronic voting. In Estonia, legally binding vote casting via Internet has been possible since 2005. Various tryouts have also taken place in Norway, Switzerland, Canada, Australia, and elsewhere (we refer interested readers to [27, 16] for good overviews).

A recent interesting newcomer in this line is Russia. During the Moscow local elections of 2019, it was possible to cast votes via the Internet. The source code of the system was opened for public scrutiny, but the accompanying documentation was rather poor. Nevertheless, serious cryptographic issues were identified in the system by Gaudry and Golonev [15].

By the 2021 parliamentary elections, two new voting systems were developed. Kaspersky Lab and the Department of Information Technologies of Moscow developed the system to conduct e-voting in Moscow [30]. Rostelecom and Waves Enterprise developed the e-voting system for six federal districts of Russia [19]. This paper concentrates on the latter, subsequently called the federal system.

As it was the case in 2019, the documentation is still poor, but the cryptographic set-up is more involved, so the primary target of this paper is to piece together what is possible to gather from various public sources about the federal protocol to enable further studies by the international community. We will provide an initial high-level analysis as the second contribution of this paper as well.

## 2 Russian Federal Remote E-voting System

An overview of the system can be obtained from the Central Election Commission's website [6]. On a high level, the protocol relies on homomorphic tallying, with the voter anonymity provided by blind signatures. The system makes significant use of a blockchain for publishing various values produced during the protocol run.

### 2.1 Protocol participants

In this section, we describe the main building blocks of the Russian remote e-voting system (Дистанционное электронное голосование, ДЭГ). The main participants of the protocol are the following:

- **Voter** (Участник ДЭГ) is a citizen of the Russian Federation who is eligible to vote and is included in the lists of e-voters based on an application submitted in electronic form. Each Voter has their personal **SNILS** (СНИЛС) [22] – individual insurance account number. The Voter uses a **Voting Device**

to participate in e-voting. The Voting Device is any device with browser and Internet access (laptop, smartphone, tablet, etc.). The voting can be done through mobile application available for Android and iOS or through browser.

- **Organiser** (Организатор) is a participant who coordinates e-voting process. They are also responsible for the generation of encryption key that is used to encrypt all the votes.
- **Internal Observer** (Внутренний Наблюдатель) is a participant who is monitoring the voting process from a dedicated room and can access individual nodes of the Blockchain component. During auditing, the Internal Observer is able to perform the following actions:
  - verify that for every Voter, to whom the blind signature was issued according to the Registrar's list, there exists a valid id_token from ESIA,
  - verify that for every Voter, to whom the blind signature was issued, there exists a transaction in the Blockchain with commitment on the Voter's SNILS code,
  - verify commitments on the Voters' SNILS codes
  - verify that the amount of cast votes is not bigger than the amount of voters to whom blind signatures were issued.
  - verify correctness of the blind signatures and the Voters' signatures published together with encrypted votes,
  - verify uniqueness of transactions (there is only one transaction for each Voter's public key).
- **External Observer** (Внешний Наблюдатель) is any user who has access to `https://stat.vybory.gov.ru`, and is able to perform the following actions:
  - verify the integrity of the Blockchain transactions,
  - verify range proofs associated to encrypted votes,
  - verify correctness of the homomorphically aggregated ciphertext,
  - verify zero-knowledge proofs of correctness of partial decryption,
  - verify if partial decryptions were aggregated correctly.

The main system components are the following.

- **Registrar** (Регистратор) consists of *Voting Portal* and *VoterList* components. It performs identification and authentication of the Voters through **ESIA** (ЕСИА) system. ESIA is the unified identification and authentication system of the Russian Federation, providing authorised access for citizens to the information contained in state information systems [20]. Additionally, the Registrar issues blind signatures to the Voters' public keys.
- **Vote collector** (Избирательный ящик) is a component that issues ballots to the Voters and collects encrypted votes. It interacts with the Blockchain to publish encrypted votes.
- **Tallier** (Учетчик) consists of *Distributed storage (Blockchain)* and *Decryptor* components. Blockchain stores all the transactions and Decryptor performs vote tallying.

### 2.2 E-voting protocol

The voting process consists of the following phases: setup, authorisation, voting, and tallying. Table 1 summarises the main cryptographic primitives used in the protocol.

**Table 1.** Main cryptographic algorithms used in Russian federal e-voting protocol

| **Functionality** | **Cryptographic scheme** |
|---|---|
| Hash function | GOST 34.11-2012 (Stribog) [26] |
| Voter anonymisation | RSA blind signature scheme [5] |
| Key sharing and encryption key generation | EC ElGamal [13] + Shamir Secret Sharing [23] |
| Encryption of the vote | EC ElGamal encryption [13] |
| Signing of the vote | GOST 34.10-2012 [25] |
| Range proofs associated with encrypted vote | Disjunctive Chaum-Pedersen proof [10] |
| Aggregation of encrypted votes | EC ElGamal encryption with additive homomorphic properties |
| Proof of correctness of decryption | Chaum-Pedersen proof [10] |
| Commitment scheme | HMAC_GOSTR3411_2012_256 [24] |

Below we give an overview of the Russian federal e-voting process. Our presentation primarily relies on the description from [8], with further details added from [7] and [9].

The Organiser and Registrar generate key pairs for GOST signature and send verification keys to the Tallier. All the messages that are sent by the Organiser and Registrar are signed with their private keys and signatures are verified by the Tallier.

**Setup phase**

- The Voter fills a digital application on the website `gosuslugi.ru` to participate in e-voting. The Voter receives a code via SMS or email to confirm their application. The Registrar compares data in each application with the Russian Federation Voter Register. Then the Registrar forms the list of eligible voters (VoterList). The Voters included in the list are excluded from the voter lists at local polling stations.
- The Organiser, in presence of Election Observer and media, generates ElGamal key pair $(S_{org}, Q_{org})$. Secret key $S_{org}$ is split into shares using Shamir Secret Sharing. Key shares are transferred to external storage media of secret key holders. Key holders are participants of the voting process selected by the Organiser (e.g. Organiser committee members, Internal Observers). $S_{org}$ is deleted from the device where it was generated.
- The Organiser receives VoterList, e-voting protocol description, ballot text, number of options in each ballot ($n$), and maximum number of options that

each Voter can select ($d$) on an external storage medium. The Organiser gives a command to generate smart contracts to the Blockchain. The Blockchain component and smart contract will be more thoroughly discussed in Section 2.3.

- The Registrar generates RSA blind signature key pair $(sk_b, pk_b)$ and sends $pk_b$ to the Organiser. The Registrar generates a commitment key $K_{com}$.
- Decryptor generates an ElGamal key pair $(S_t, Q_t)$ and sends the public key $Q_t$ to the Organiser. The key pair is generated in hardware security module (HSM) and private key $S_t$ never leaves HSM.
- The Organiser uploads identifier of elections $votingID$, starting time of receiving ballots, a hash of the ballot text, $n$, $d$ and $pk_b$ to the Blockchain. Based on this information, the Blockchain smart contracts are generated.
- The Registrar computes commitments on the Voters' SNILS codes as $com_i = HMAC(K_{com}, SNILS||votingID)$ and publishes these into the Blockchain. The Blockchain smart contracts are updated by adding received commitments.
- The Organiser constructs final encryption key $Q_f = H(Q_t||Q_{org}) \cdot Q_{org} + H(Q_{org}||Q_t) \cdot Q_t$ and uploads $Q_{org}, Q_t, Q_f$ to the Blockchain. The Registrar receives $Q_f$ from the Blockchain.

The general process of the setup phase is presented in Figure 1.

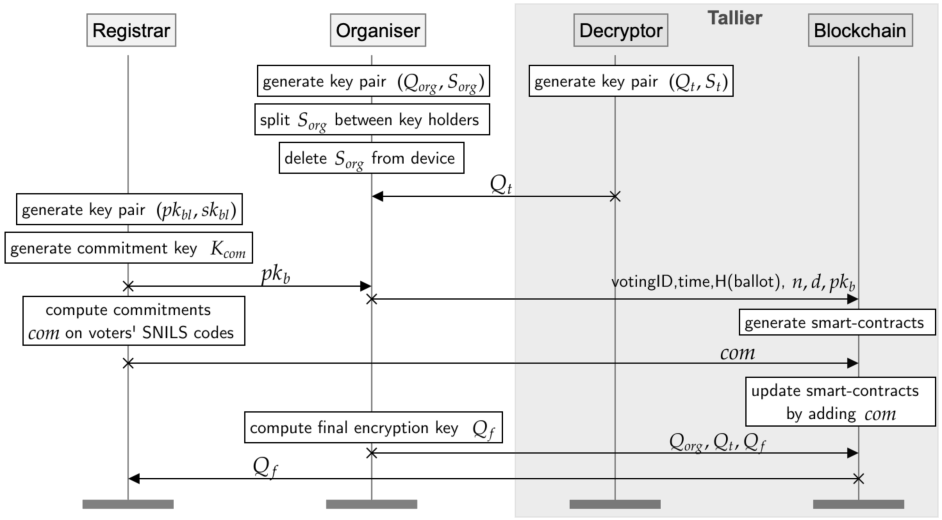


**Fig. 1.** Setup phase

### Authorisation phase

– The Voter authenticates to the e-voting portal through the ESIA system. The Registrar receives the id_token and the information about the Voter from ESIA. The Registrar checks the eligibility of the Voter by their SNILS code. The Voter receives SMS or email with an authorisation code that they enter into the e-voting portal.
– The Voting Device generates a key pair $(sk_v, pk_v)$ for the GOST signature. The Voting Device interacts with the Registrar to receive RSA blind signature $s$ on the Voter's masked public key.
– The Registrar records the fact of blind signature issuance by updating its internal database and publishing $com, s$ into the Blockchain, $com$ is the commitment of the Voter's SNILS code.
– The Voting Device removes the mask from the signature $s$. The resulting value $\sigma_b$ is a valid RSA signature on the Voter's public key.

Authorisation process is depicted in Figure 2.

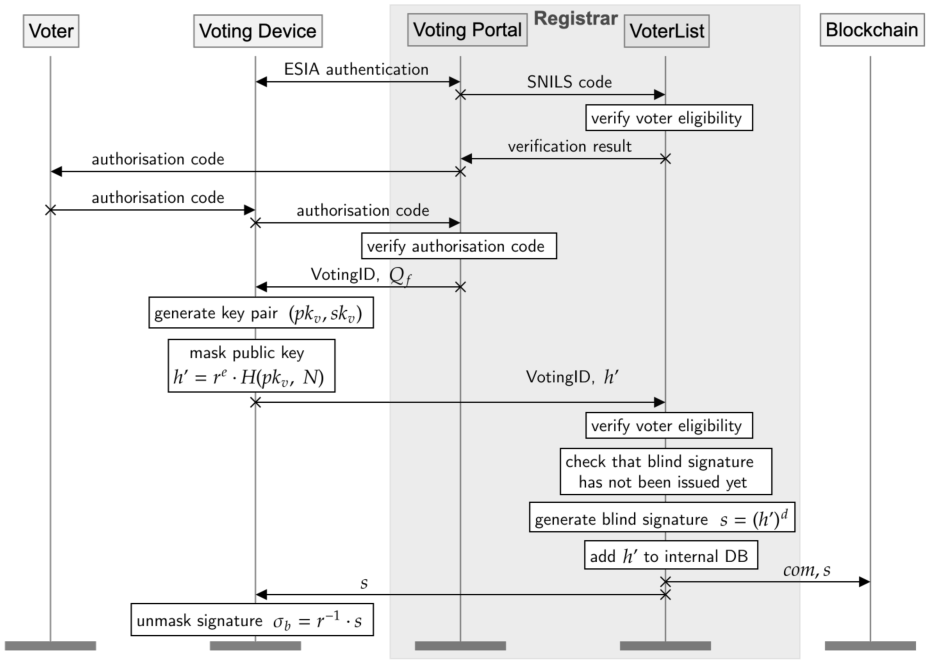


**Fig. 2.** Authorisation phase

### Voting phase

- The Voter is redirected to the anonymous zone of the voting portal (Vote Collector). The Vote Collector receives $pk_v$ and checks that for this public key a ballot has not been issued before.
- The Voter is presented with a ballot in digital form and the Voter makes their choice by selecting the preferred option.
- Each ballot is represented as a bitstring of length $n$: $v = v_0 v_1 \dots v_{n-1}$. The initial value of each option on the ballot is zero, the option chosen by the Voter has value one. Each option ($v_i = 0$ or $v_i = 1$) is encrypted separately using ElGamal encryption as $(c_v)_i = Enc(v_i, Q_f)$, for $i \in \{0, \dots, n-1\}$.
- The Voting Device generates several range proofs

$$(\pi_v)_i = genProof(Q_f, rng, v_i, r_v, (c_v)_i) \ ,$$

  where $r_v$ is a random value, $rng$ is the range of possible plaintext values and $i \in \{0, \dots, n-1\}$. For each ciphertext, the Voting Device generates proof that the encrypted value is either 0 or 1. Additionally, the Voting Device generates proof that the sum of all values does not exceed the bound $d$ (as the Voter can choose up to $d$ options).
- Finally, the Voting Device creates a signature $\sigma_v = Sign((c_v, \pi_v, pk_v, \sigma_b), sk_v)$, where $c_v$ corresponds to all created ciphertexts and $\pi_v$ corresponds to all generated range proofs. The Voter sends the transaction $t_v = \{c_v, \pi_v, pk_v, \sigma_b, \sigma_v\}$ to the Vote Collector.
- The Vote Collector verifies the signature $\sigma_b$ and uniqueness of the transaction containing $pk_v$. The Vote Collector updates its internal database by adding $pk_v$ and sends the transaction $t_v$ to the Blockchain.
- The Blockchain smart contract verifies the signatures $\sigma_b$ and $\sigma_v$, and publishes the transaction.

The ballot generation process and voting depicted in Figures 3 and 4, respectively.

**Tallying phase**

- The Organiser requests the Registrar to stop authenticating new Voters and the Blockchain to stop accepting new voting transactions.
- The Organiser reconstructs their private key $S_{org}$ from the shares and publishes $S_{org}$ into the Blockchain.
- The Decryptor receives all the voting transactions from the Blockchain and verifies range proofs $(\pi_v)_i$ for each transaction. The Decryptor sums up verified encrypted votes separately for each option from the ballot as $sum_i = \sum_{v=0}^{V-1} (c_v)_i = (R_i, C_i)$, where $i \in \{0, \dots, n-1\}$ and $V$ is the total number of cast votes.
- Decryption of aggregated ciphertexts $sum_i$ is performed in two steps. Firstly, the Decryptor computes partial decryptions as $(R_i)_t = S_t \cdot R_i$. Next, the Decryptor generates proofs of correctness of partial decryptions $(\pi_i)_t = genProof(S_t, (R_i)_t, R_i, Q_t, P)$ and publishes $(R_i)_t, (\pi_i)_t$ ($i \in \{0, \dots, n-1\}$) into the Blockchain.

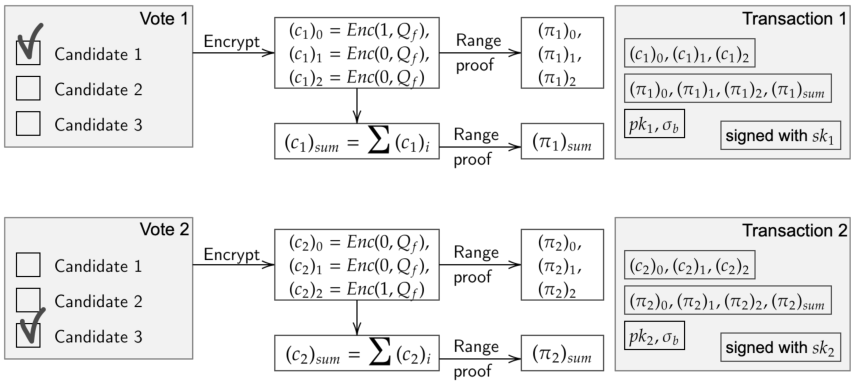


**Fig. 3.** Ballot encryption

- The Decryptor receives Organiser's private key $S_{org}$ and verifies that it corresponds to the previously published public key $Q_{org}$.
- Finally, the Decryptor performs the final decryption of aggregated votes using $S_{org}$ as

$$M_i = C_i - H(Q_t||Q_{org}) \cdot S_{org} \cdot R_i - H(Q_{org}||Q_t) \cdot (R_i)_t$$

and publishes the transaction $\{(R_i, C_i), M_i\}$, where $i \in \{0, \ldots, n-1\}$.

Figures 5 and 6 depict the processes of decryption and tallying, respectively.

Only the Organiser's private key is published into the Blockchain. Thus it is not possible to decrypt transactions containing individual votes using publicly available information. It means that the Voter cannot verify if the vote published into the Blockchain corresponds to their original choice.

There are different descriptions of the setup phase in two official documents [7, 8]. The main difference is in which component is responsible for the generation of the final encryption key $Q_f$ (Organiser or Tallier). Additionally, there is no verification that the final key $Q_f$ was generated as it was supposed to be generated. The documents differ also in the tallying phase. It is not clear if the Decryptor needs to compute partial decryptions and corresponding zero-knowledge proofs with both keys $S_{org}$ and $S_t$ or with the Tallier's key $S_t$ only.

Let us briefly consider two possible setups. In the first case,

- the Tallier (Decryptor) is responsible for the final key generation,
- there is no verification that the final key $Q_f$ was generated correctly, and
- the Tallier (Decryptor) computes and publishes into the Blockchain partial decryption only using its private key $S_t$.

In this case, if an adversary corrupts the Tallier, they can compute the final key $Q_f$ without including the Organiser's share $Q_{org}$ to it. This would allow the

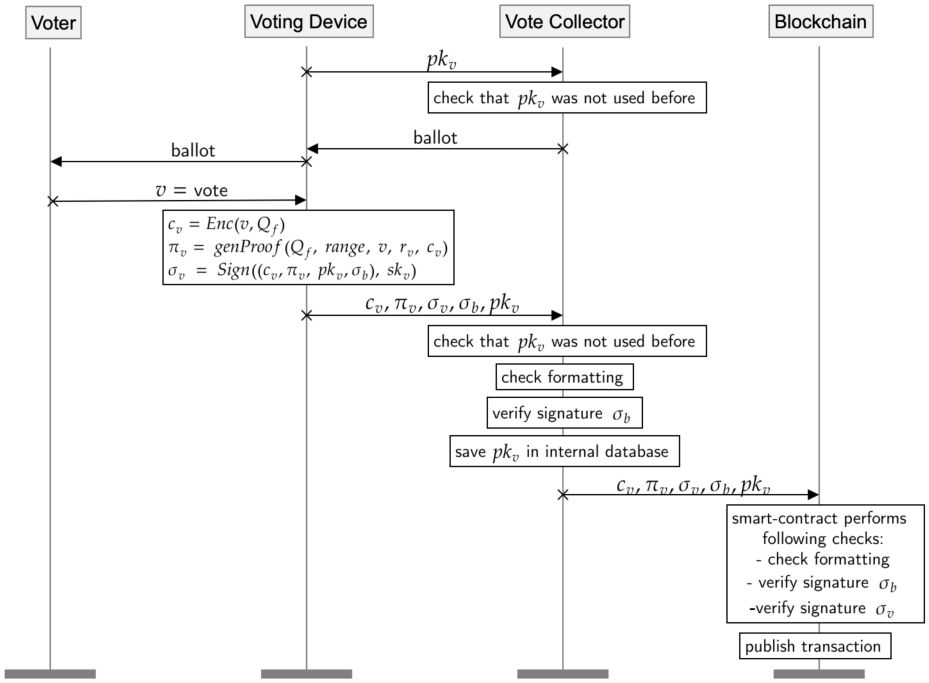


**Fig. 4.** Voting phase

adversary to decrypt individual votes during the voting phase. Moreover, as there are no zero-knowledge proofs of decryption correctness using the Organiser's key, this would remain unnoticed.

In the case of the second possible setup,

- the Organiser is responsible for the final key generation,
- there is no verification that the final key $Q_f$ was generated correctly, and
- the Tallier (Decryptor) computes and publishes into the Blockchain partial decryption only using its private key $S_t$.

In this case, if an adversary corrupts both Tallier and Organiser, they can compute the final key $Q_f$ without including Organiser's share $Q_{org}$ to it. In this case, there will be no need to reconstruct $S_{org}$ from the shares to decrypt individual votes during the voting phase. Again, as there are no zero-knowledge proofs of decryption correctness using the Organiser's key, this would remain unnoticed.

### 2.3 Usage of blockchain in the protocol

The Blockchain platform that was used for e-voting was developed by Waves Enterprise [29]. It uses Crash Fault Tolerance (CFT) consensus algorithm that

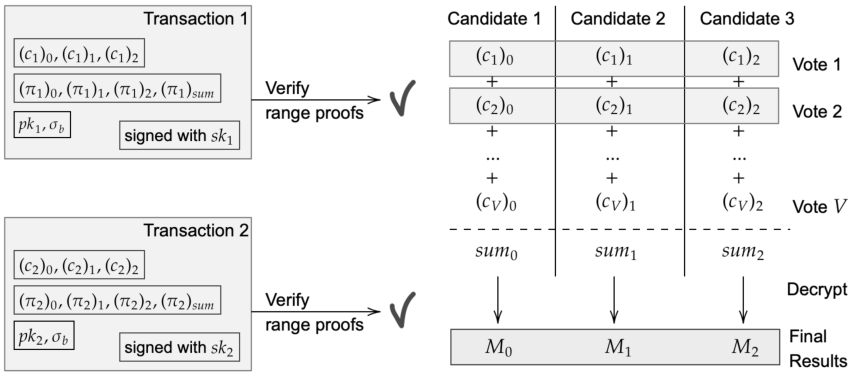


**Fig. 5.** Decryption and tallying of results

is based on Proof of Authority (PoA) consensus [29]. The Blockchain platform supports the development and usage of Turing complete smart contracts. Smart contract used in the e-voting process performs the following functions:

- storing the rules of the voting process and the list of participants,
- registering information, obtained during the setup phase, and
- verification and storage of the cast votes and voting results [28].

There are four data processing centers that run the Blockchain nodes, these centers are managed by Rostelecom and the Registrar.

## 3 Protocol analysis

In this section, we analyse the security properties of the Russian federal e-voting protocol. Our analysis is based on Neumann's approach [21] aiming to identify the minimal subsets of the parties who can breach some of the security properties. We target the following properties also adapted from Neumann [21].

- *Vote Integrity* – the voting system must ensure that each vote has not been tampered with during the voting process, and is correctly included in the election result.
- *Vote Secrecy* – the voting system must ensure that it is impossible to link the content of the cast vote to the Voter's identity.
- *Voter Eligibility and Uniformity* – the voting system must ensure that only votes of eligible voters are accepted, these votes are accepted only once and with equal weight.
- *Data Access Protection* – the voting system must prevent unauthorised users from accessing the Voter's personal data.

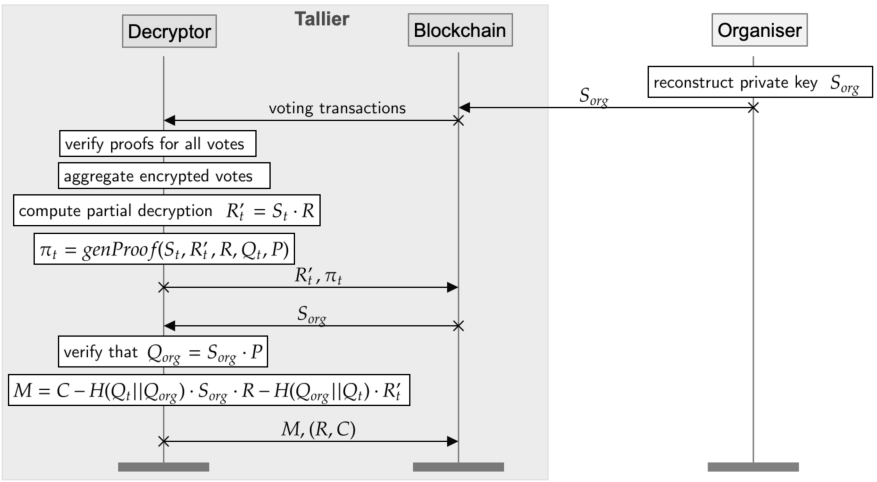


**Fig. 6.** Tallying phase

### 3.1 Vote Integrity

*Vote Collector* The corrupt Vote Collector can drop an incoming transaction with a vote and not transfer it further to the Blockchain. This can only be noticed by the Voter and the Organiser. If the Voter has saved their public key or transaction ID, they can later check if their vote was actually added to the Blockchain. The Organiser may notice the Vote Collector's malicious activity by checking the logs of the Vote Collector.

*Tallier* If an adversary compromises more than half of the Blockchain nodes, they can modify encrypted votes during the voting phase [28]. However, zero-knowledge proofs associated with these votes will not pass verification. Therefore, the modified votes will not be counted in the final result.

*Voting Device* If an adversary corrupts the Voting Device, they may change the Voter's choice before it gets encrypted and sent to the Vote Collector. There's no guarantee for the Voter that the voting page is indeed provided by the Registrar and not by the adversary [17]. Additionally, the system does not provide individual vote verifiability, the Voter will not be able to tell if the vote published into the Blockchain corresponds to their real choice.

*Voter's account* ESIA system supports two-factor authentication through SMS-code and email codes (for those who received citizenship in a simplified manner [9]). If an adversary gets access to the Voter's ESIA credentials, they will be able to execute several attacks.

– An adversary can use the Voter's credentials to log in to `gosuslugi.ru`. Next, the adversary adds their phone number to the account and changes the Voter's password. The adversary can register the Voter for e-voting and

cast the vote on behalf of the Voter. This attack remains unnoticed if the Voter did not have the intention to participate in elections.
- Secondly, an adversary may target the Voters who received citizenship in a simplified manner. The adversary can get hold of the Voter's email password and execute the previous attack even without changing the phone number.
- According to the documentation [9], if the Voter enters an incorrect authorisation code during the authorisation phase, the Registrar provides them with a new code (no more than once a minute). The number of attempts to enter the code is limited by the end time of e-voting. The adversary can get access to the Voter's account and make attempts to guess the code every minute until the end of elections. The probability of success for 5 digit code is around 3%. However, this attack will be noticed by the Voter, who has physical access to their phone.

### 3.2 Vote Secrecy

*Voting Device* The Voter receives a ballot through the e-voting portal or using a mobile application on their Voting Device. If the Voting Device gets compromised, the Voter's choice may get leaked to the adversary and linked to the Voter's identity. There are several ways how the adversary can compromise the Voter's device described in [17].

*Registrar and Vote Collector* During the authorisation phase, the corrupt Registrar can save the Voter's IP address and browser metadata. Later, during the voting phase, the Vote Collector can also record the same information. As a result, the Registrar knows the Voter's identity, IP address, browser, and device details. The Vote Collector knows the Voter's encrypted vote and IP address with metadata. Comparing this information, they can link an encrypted vote to the Voter's identity. However, to be able to decrypt the individual vote, the adversary would need to compromise the Organiser and Decryptor (HSM module).

### 3.3 Voter Eligibility and Uniformity

The eligibility of the Voter is checked by the Registrar during the authentication process and the Vote Collector during the voting phase. It is not possible to verify using publicly available data if the vote published into the Blockchain was cast by an eligible voter.

*Organiser and Voter* While preparing the final VoterList, the corrupt Organiser may not exclude the Voter who registered to participate in e-voting from the list of voters on the polling station. This would allow the Voter to cast their vote twice. This attack remains unnoticed as the VoterList for e-voting and VoterList at the polling stations are not cross-checked during the audit phase.

*Organiser, Registrar and Voter* An ineligible Voter can fill an application on the website `gosuslugi.ru`. The corrupt Registrar can approve the application and the corrupt Organiser can add the Voter to the final VoterList. The ineligible Voter could later cast their vote using the e-voting system.

*Organiser, Registrar, ESIA, Vote Collector and Election Observer*. The corrupt Registrar can issue blind signatures for illegitimate public keys and corrupt Vote Collector adds illegitimate votes to the system. Since the Organiser is also corrupt, this will remain undetected. Additionally, the adversary should ensure that for every illegitimate vote there exists an id_token issued by ESIA (this is checked in the audit phase). Alternatively, Election Observer, who performs the audit, may be corrupted by the adversary. In this case, there is no need to have the tokens issued by ESIA.

### 3.4 Data Access Protection

During the voting process, the following components process Voter's personal data:

- *Voting Device* – stores information about the Voter's identity.
- *Registrar* – receives the Voter's personal data from ESIA during the authentication process. This data includes first and last name, date of birth, SNILS code, phone number, and information about the document (passport).
- *Organiser* – stores the list of Voters, which includes personal information.

Therefore, if the adversary corrupts at least one of the components listed above, it results in a violation of data access protection.

## 4 Conclusions and Future Work

In this work, we have presented an overview of the Russian federal e-voting system. We based our analysis on the materials published by the Central Election Commission of Russia. However, this documentation is incomplete and misleading, making it hard to follow and analyse. Additionally, we used information published by the developers of the voting system components, public presentations, and media articles. As a result, we managed to compile the workflow of the federal e-voting system and provide an initial analysis of the system.

We found that the main weakness of the system lays in the authentication process. The usage of a password-based authentication system (ESIA) leads to a higher chance of different attacks against vote integrity. Furthermore, the attack surface increases as ESIA is used to authenticate users to more than 1000 IT systems [18].

Secondly, the analysis showed that the adversary will be able to break three out of four security properties of the system by compromising the Voting Device.

Additionally, the system uses a non-standard approach for generating the ElGamal encryption key. In the documentation [6] it has not been specified if this approach is invented by the authors of the e-voting protocol or based on the prior work. The better practice is to use more established and better understood cryptographic techniques such as threshold ElGamal [11] instead of introducing a new key sharing technique.

We consider this work as a starting point for future analysis of the Russian federal e-voting system. Security analysis of cryptographic primitives, code audit, and analysis of election statistics remain interesting targets for future work. The e-voting system used in Moscow featured a completely different cryptographic protocol, so future research is needed to study it as well.

**Acknowledgements** This paper has been supported by the Estonian Research Council under the grant number PRG920.

## References

1. Appel, A.W., Ginsburg, M., Hursti, H., Kernighan, B.W., Richards, C.D., Tan, G.: Insecurities and inaccuracies of the Sequoia AVC Advantage 9.00 H DRE voting machine (2008)
2. Appel, A.W., Ginsburg, M., Hursti, H., Kernighan, B.W., Richards, C.D., Tan, G., Venetis, P.: The New Jersey Voting-machine Lawsuit and the AVC Advantage DRE Voting Machine. In: 2009 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections, EVT/WOTE '09. USENIX Association (2009)
3. Aviv, A.J., Cerný, P., Clark, S., Cronin, E., Shah, G., Sherr, M., Blaze, M.: Security Evaluation of ES&S Voting Machines and Election Management System. In: 2008 USENIX/ACCURATE Electronic Voting Workshop, EVT 2008 (2008)
4. Bannet, J., Price, D.W., Rudys, A., Singer, J., Wallach, D.S.: Hack-a-vote: Security issues with electronic voting systems. IEEE Security & Privacy **2**(1), 32–37 (2004)
5. Bellare, M., Namprempre, C., Pointcheval, D., Semanko, M.: The one-more-rsa-inversion problems and the security of chaum's blind signature scheme. Cryptology ePrint Archive, Report 2001/002 (2001), `https://ia.cr/2001/002`
6. Central Election Commission of Russia: Дистанционное электронное голосование. `https://vybory.gov.ru/materials`, accessed: 2021-10-21
7. Central Election Commission of Russia: Описание ПТК ДЭГ. `https://vybory.gov.ru/landing/materials/deg2021-docs/deg2021-description.pdf`, accessed: 2021-10-20
8. Central Election Commission of Russia: Описание протокола ДЭГ к выборам, голосование на которых состоится 17, 18 и 19 сентября 2021 г. `https://vybory.gov.ru/landing/materials/7/deg2021_protocol.pdf`, accessed: 2021-10-20
9. Central Election Commission of Russia: Порядок дистанционного электронного голосования на выборах, назначенных на 19 сентября 2021 года. `http://cikrf.ru/upload/decree-of-cec/26-225-8-pril.docx` (July 2021), accessed: 2021-10-20
10. Chaum, D., Pedersen, T.P.: Wallet databases with observers. In: Brickell, E.F. (ed.) Advances in Cryptology — CRYPTO' 92. pp. 89–105. Springer (1993)
11. Desmedt, Y., Frankel, Y.: Threshold cryptosystems. In: Brassard, G. (ed.) Advances in Cryptology — CRYPTO' 89 Proceedings. pp. 307–315. Springer New York, New York, NY (1990)
12. Dill, D.L., Schneier, B., Simons, B.: Voting and technology: who gets to count your vote? Commun. ACM **46**(8), 29–31 (2003)
13. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Advances in Cryptology. pp. 10–18. Springer (1985)

14. Feldman, A.J., Halderman, J.A., Felten, E.W.: Security Analysis of the Diebold AccuVote-TS Voting Machine. In: 2007 USENIX/ACCURATE Electronic Voting Technology Workshop, EVT'07 (2007)
15. Gaudry, P., Golovnev, A.: Breaking the Encryption Scheme of the Moscow Internet Voting System. In: Bonneau, J., Heninger, N. (eds.) Financial Cryptography and Data Security - 24th International Conference, FC 2020, Kota Kinabalu, Malaysia, February 10-14, 2020 Revised Selected Papers. Lecture Notes in Computer Science, vol. 12059, pp. 32–49. Springer (2020)
16. Gibson, J.P., Krimmer, R., Teague, V., Pomares, J.: A review of E-voting: the past, present and future. Ann. des Télécommunications **71**(7-8), 279–286 (2016)
17. Heiberg, S., Krips, K., Willemson, J.: Mobile voting – still too risky? In: Financial Cryptography and Data Security. FC 2021 International Workshops. pp. 263–278. Springer (2021)
18. Identity Blitz: Russian e-government system of trusted identities. `https://identityblitz.com/?portfolio=russian-e-government-system-of-trusted-identities`, accessed: 2021-10-28
19. Mikhail Tetkin, RBC: Ростелеком» разработает систему голосования на блокчейне по заказу ЦИК. `https://www.rbc.ru/crypto/news/5f3d04e69a79475a99a7526d` (August 2020), accessed: 2021-10-21
20. Ministry of Digital Development, Communications and Mass Media of the Russian Federation: Единая система идентификации и аутентификации (ЕСИА). `https://digital.gov.ru/ru/activity/directions/13/`, accessed: 2021-10-20
21. Neumann, S.: Evaluation and Improvement of Internet Voting Schemes Based on Legally-Founded Security Requirements. Ph.D. thesis, Technische Universität Darmstadt (03 2016)
22. Pension Fund of the Russian Federation: СНИЛС, как получить, заменить и восстановить. `https://pfr.gov.ru/spec/infographics1/snils.html`, accessed: 2021-10-20
23. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (November 1979)
24. Smyshlyaev, S.V., Alekseev, E., Oshkin, I., Popov, V., Leontiev, S., Podobaev, V., Belyavsky, D.: Guidelines on the Cryptographic Algorithms to Accompany the Usage of Standards GOST R 34.10-2012 and GOST R 34.11-2012. RFC 7836 (Mar 2016), `https://rfc-editor.org/rfc/rfc7836.txt`
25. V. Dolmatov, A.D.: Gost r 34.10-2012: Digital signature algorithm. `https://datatracker.ietf.org/doc/html/rfc7091` (December 2013), accessed: 2021-10-20
26. V. Dolmatov, A.D.: Gost r 34.11-2012: Hash function. `https://datatracker.ietf.org/doc/html/rfc6986` (August 2013), accessed: 2021-10-20
27. Vegas, C., Barrat, J.: Overview of current state of E-voting worldwide. In: Real-World Electronic Voting, pp. 67–92. Auerbach Publications (2016)
28. Waves Enterprise: Technical description of the waves enterprise voting. `https://docs.we.vote/en/votingdocs.pdf`, accessed: 2021-10-20
29. Waves Enterprise: Семинар «Технологии блокчейн и криптозащиты в системе ДЭГ. `https://wavesenterprise.com/ru/media/seminar-tekhnologii-blokchejn-i-kriptozashchity-v-sisteme-deg`, accessed: 2021-10-20
30. Департамент города Москвы по конкурентной политике: Протокол подведения итогов открытого конкурса в электронной форме от 12.05.2021 №ППИ1.

https://zakupki.gov.ru/epz/order/notice/ok504/view/supplier-results.html?regNumber=0173200001421000490 (May 2021), accessed: 2021-10-21