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

Jelizaveta Vakarjuk*†, Nikita Snetkov*†‡, Jan Willemson*
*Cybernetica AS, Mäealuse 2/1, 12618 Tallinn, Estonia
{jelizaveta.vakarjuk, nikita.snetkov, jan.willemson}@cyber.ee
†STACC OÜ, Narva mnt 20, 51009 Tartu, Estonia
‡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 the 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.

I. Introduction

The usage of electronic means to support the process of voting has been a subject of extensive research and debate. 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 [1]–[6].

Voting over the 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 efficiently 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 [7], [8] 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 [9].

By the 2021 parliamentary elections, two new voting systems were deployed. Kaspersky Lab and the Department of Information Technologies of Moscow developed a system to conduct e-voting in Moscow [10]. Rostelecom and Waves Enterprise developed an e-voting system for six federal districts of Russia [11]. 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 system to enable further studies by the international community. As the second contribution of this paper as well, we will provide an initial high-level analysis.

II. Russian Federal Remote E-voting System

An overview of the system can be obtained from the Central Election Commission's website [12]. 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 public blockchain for publishing various values produced during the protocol run.

A. Participants

In this section, we introduce the main participants of the Russian federal e-voting protocol and their roles in the system.

Voter is a citizen of the Russian Federation who is eligible to vote and is included in the lists of e-voters (VoterList) based on an application submitted in electronic form through gosuslugi.ru. gosuslugi.ru is a web portal that provides access to information about state and municipal services in the Russian Federation. The Voter must have verified their gosuslugi.ru account, and be eligible to vote to register as an e-voting participant. The Voters included in VoterList are excluded from the lists at the local polling stations. Each Voter has their personal SNILS [13] – an individual insurance account number. The Voter uses a Voting Device to participate in e-voting. Voting Device is any device with a browser and Internet access (laptop, smartphone, tablet, etc.). Voting can be performed through gosuslugi.ru mobile application available for Android and iOS, or through a browser. During the authorisation phase, the Voting Device generates a key pair for the GOST signature scheme (see Table I).

Organiser is a participant who coordinates the e-voting process. It is also responsible for the generation of Organiser's key pair and the final encryption key that is used to encrypt all the votes (see Figure 1).

Internal Observer is a participant who is monitoring the voting process from a dedicated room and can access individual nodes of the Blockchain component. Additionally, Internal Observer performs the auditing process.

External Observer is any user who has access to https://stat.vybory.gov.ru. Unfortunately, one month after the elections the website became inaccessible.

Election Observer refers further in the text to both Internal and External Observers.

Key holders are participants of the voting process selected by the Organiser (e.g. Organiser committee members, Internal Observers) who hold shares of the Organiser's secret key.

Registrar consists of the Voting Portal and VoterList components. It performs identification and authentication of the Voters through ESIA 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 [14]. Additionally, the Registrar issues blind signatures to the Voters' public keys.

Vote collector is a separate component that allows the Voter to cast their vote while maintaining the secrecy of their vote. This component issues ballots to the Voters and collects encrypted votes. It interacts with the Blockchain to publish encrypted votes.

Tallier consists of the Distributed storage (Blockchain) and Decryptor components. Blockchain stores all the voting transactions and published keys. Decryptor has a hardware security module (HSM), where the Tallier's key pair is generated and the vote tallying is performed (see Figure 1).

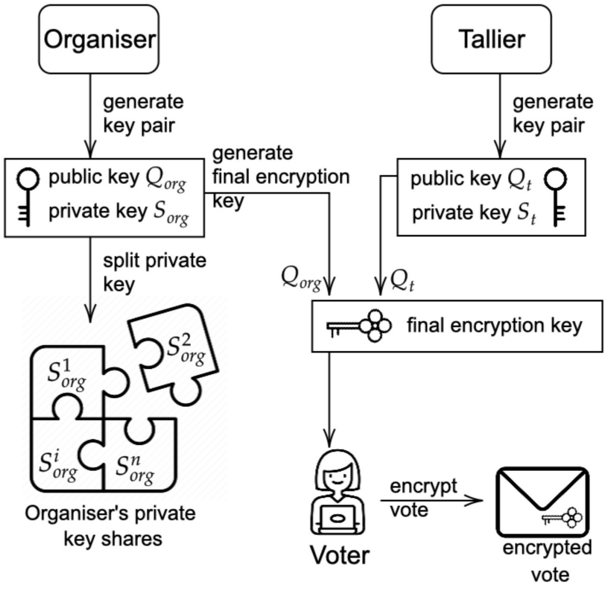

Fig. 1. Relationships between the encryption keys

B. Usage of blockchain in the protocol

The Blockchain platform that was used for e-voting was developed by Waves Enterprise [15]. It uses the Crash Fault Tolerance (CFT) consensus algorithm that is based on Proof of Authority (PoA) consensus [15]. 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 [16].

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

C. Protocol

Below, we give an overview of the Russian federal e-voting protocol. Our presentation primarily relies on the description from [17], with further details added from [18] and [19]. Table I summarises the main cryptographic primitives used in the protocol.

1) Setup phase:

- The Organiser and Registrar generate key pairs for the GOST signature scheme and send public keys to the Tallier. All the messages that are sent by the Organiser and Registrar are signed with their secret keys and signatures are verified by the Tallier.
- The Registrar generates an RSA blind signature key pair (sk_b, pk_b), sends pk_b to the Organiser and generates a commitment key K_{com}.

TABLE I
Main cryptographic protocols and algorithms used in Russian federal e-voting protocol

Functionality	Cryptographic scheme
Hash function	GOST 34.11-2012 (Stribog) [20]
Voter anonymisation	RSA blind signature scheme [21]
Key sharing and encryption key generation	Elliptic Curve ElGamal [22] + Shamir Secret Sharing [23]
Encryption of the vote	Elliptic Curve ElGamal encryption [22]
Signing of the vote	GOST 34.10-2012 [24]
Range proofs associated with encrypted vote	Disjunctive Chaum-Pedersen proof [25]
Aggregation of encrypted votes	EC ElGamal encryption with additive homomorphic properties
Proof of correctness of decryption	Chaum-Pedersen proof [25]
Commitment scheme	HMAC_GOSTR3411_2012_256 [26]

- The Organiser, in presence of Election Observers and media, generates an ElGamal key pair (S_{org}, Q_{org}). The secret key S_{org} is split into shares using Shamir Secret Sharing. Key shares are transferred to external storage media of secret key holders. S_{org} is deleted from the device, where it was generated.
- Decryptor generates an ElGamal key pair (S_t, Q_t) and sends Q_t to the Organiser.
- The Organiser uploads identifier of elections $(votingID)$, starting time of receiving ballots, a hash of the ballot text, a number of options in each ballot (n), maximum number of options that each Voter can select (d), and pk_b to the Blockchain. Based on this information, the Blockchain smart contracts are generated. The Organiser sends the VoterList to the Registrar.
- The Registrar computes commitments
$$com = HMAC(K_{com}, SNILS||votingID)$$
on the Voters' SNILS codes from the VoterList and uploads these into the Blockchain. The Blockchain smart contracts are updated by adding the received commitments.
- The Organiser constructs the final encryption key
$$Q_f = H(Q_t||Q_{org}) \cdot Q_{org} + H(Q_{org}||Q_t) \cdot Q_t \qquad (1)$$
and uploads Q_{org}, Q_t, Q_f to the Blockchain. The Registrar receives Q_f from the Blockchain. The final encryption key Q_f is constructed from the Organiser's and the Tallier's public keys, therefore to decrypt the votes, both secret keys are needed. As the Organiser's secret key is split until the tallying phase, it prevents the Organiser and the Tallier from learning intermediate results.

2) Authorisation phase:

- The Voter authenticates to the e-voting portal through the ESIA system. The Registrar receives the signed id_token and the information about the Voter from ESIA. The Registrar checks the eligibility of the Voter by their SNILS code. The Registrar sends an SMS or email with an authorisation code to the Voter, and the Voter enters it into the e-voting portal.
- The Voting Device generates a key pair (sk_v, pk_v) for the GOST signature scheme. The Voting Device interacts with the Registrar to receive RSA blind signature s on the Voter's masked public key.
- The Registrar updates its VoterList by recording id_token, com, s issued for the Voter, and publishes (com, s) into the Blockchain. The Registrar sends votingID and Q_f to the Voting Device.
- The Voting Device removes the mask from the signature s. The resulting value σ_b is a valid RSA signature on the Voter's public key.

3) Voting phase:

- The Voter is redirected to the anonymous zone of the voting portal (Vote Collector). The Voter is presented with a ballot in digital form and the Voter makes their choice by selecting the preferred option.
- Each ballot b is represented as a bitstring of length n. The initial value of each option on the ballot is zero, the option chosen by the Voter changes the value to one. Each option ($b_i = 0$ or $b_i = 1$) is encrypted separately using ElGamal encryption $c_i = Enc(b_i, Q_f)$, for $i \in \{0, \ldots, n-1\}$.
- For each ciphertext, the Voting Device generates a proof that the encrypted value is either 0 or 1. Additionally, the Voting Device generates a 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 prepares the transaction consisting of all created ciphertexts and corresponding proofs, pk_v and σ_b. The Voting Device signs this transaction with sk_v to receive signature σ_v, and sends the signed transaction to the Vote Collector.
- The Vote Collector verifies the cast ballot for well-formedness, the signature σ_b and uniqueness of the transaction containing pk_v. The Vote Collector adds pk_v to its internal database and uploads the transaction to the Blockchain.
- The Blockchain smart contract verifies ballot well-formedness, the signatures σ_b and σ_v, and publishes the transaction.

The process of generating the voting transaction is depicted in Figure 2.

After casting a vote, the Voter can check if their transaction was added to the Blockchain through the

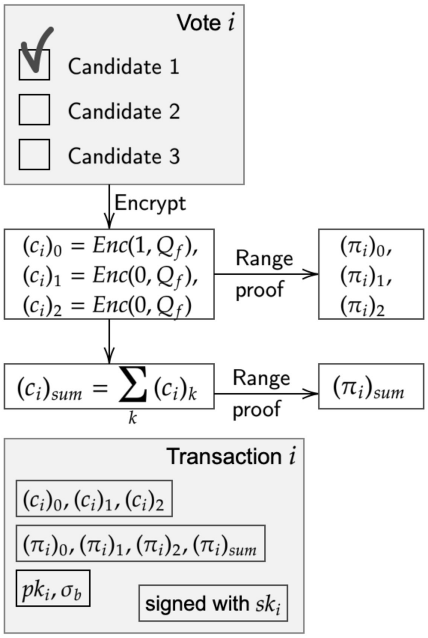

Fig. 2. Voting transaction

https://stat.vybory.gov.ru portal using their public key pk_v.

4) 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 secret key S_{org} from the shares.
- The Decryptor receives all the voting transactions from the Blockchain and verifies range proofs for each transaction. The Decryptor aggregates verified encrypted votes separately for each option from the ballot as $sum_i = \sum_{v=1}^{V} (c_v)_i = (R_i, C_i)$, where $i \in \{0, \ldots, n-1\}$ and V is the total number of cast votes.
- The 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 and uploads all partial decryptions with the corresponding proofs of decryption correctness into the Blockchain.
- The Organiser uploads S_{org} into the Blockchain. The Decryptor receives 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 $(R_i, C_i), M_i$ $(i \in \{0, \ldots, n-1\})$.

The tallying phase is depicted in Figure 3.

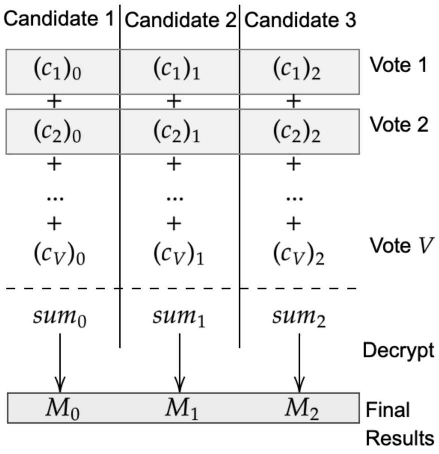

Fig. 3. Tallying phase

Only the Organiser's secret key is published. Thus it is not possible to decrypt the 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.

5) Audit

: 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 and a transaction in the Blockchain with a commitment on the Voter's SNILS code,
- verify commitments on the Voters' SNILS codes,
- verify that the number of cast votes is not bigger than the number of voters to whom blind signatures were issued,
- verify the correctness of the blind signatures and the Voters' signatures from the voting transaction,
- verify that there is only one transaction for each Voter's public key.

III. Security analysis

In this section, we analyse some security aspects of the Russian federal e-voting system. We examine the following properties.

- Cast as Intended verifiability: The Voter should be able to verify that their intent was correctly interpreted [27].
- Recorded as Cast verifiability: The Voter should be able to verify that their vote was successfully recorded without alterations [27].

- Universal verifiability: Everyone should be able to verify that the final tally is correctly calculated from the published votes [27].
- Eligibility verifiability: The system should ensure that only eligible voters are allowed to cast a vote. An adversary should not be able to undetectably add votes on behalf of the voter who did not actually vote [27].
- Coercion resistance: The Voter should be able to cast a vote that reflects their actual choice even in the presence of a coercer during the voting period [28].
- Receipt-freeness: The Voter should not be able to produce a proof to a coercer that they voted in a particular manner [28].
- Vote secrecy: It should be impossible to link the content of the cast vote to the Voter's identity [29].
- Fairness: It should be impossible to calculate the intermediate results of an election before the tallying phase has ended [29].
- Dispute resolution: If the Voter notices malicious behavior of the voting system, they should be able to prove it [27].

Cast as intended verifiability is not satisfied. The Voting scheme has no mechanism that enables the Voter to verify that their vote was interpreted correctly after marking their choice in the digital ballot.

Recorded as cast verifiability is not satisfied. The Voter can verify that their voting transaction was published into the Blockchain, but they cannot verify that the vote has not been altered. The Voter cannot decrypt their voting transaction from the Blockchain. Only one part of the secret key is published after tallying phase, therefore it is not possible to decrypt single votes.

We conclude that the scheme does not provide any form of individual verifiability. Thus, the Voter is unable to detect if the integrity of their vote has been violated. If the Voting Device is dishonest, instead of encrypting the Voter's true intent, it may create an encryption of the vote for any other option from the ballot. The modified transaction will be then signed and sent to the Vote Collector. The only thing that can be verified by the Voter in the protocol is that their voting transaction has not been dropped during the voting phase.

Universal verifiability is satisfied. The Tallier computes partial decryptions using their secret key (in HSM) and generates a proof of decryption correctness that can be verified by Election Observers. The Organiser's secret key share S_{org} is published into the Blockchain together with partial decryptions. Therefore, Election Observers can verify the correctness of the tally using public information from the Blockchain. If the Tallier is corrupt by an adversary, the Tallier is unable to provide valid proof for the manipulated tally.

Eligibility verifiability is partially satisfied. ESIA system supports two-factor authentication through SMS-code and email codes (for those who received citizenship in a simplified manner [19]). 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 a vote on behalf of the Voter. This attack remains unnoticed if the Voter did not have the intention to participate in elections. An adversary can conduct a similar attack targeting the Voters who received citizenship in a simplified manner.

According to [19], 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 the voting phase. The adversary can get access to the Voter's account and make attempts to guess the code every minute until the end of the elections. The probability of successful attack for 5 digit code is around 3%. However, this attack will be noticed by the Voter, who has physical access to their phone.

The Internal Observers can verify the correctness of the Registrar's VoterList, by verifying id_tokens issued by ESIA and commitments on the SNILS codes. Additionally, the Internal Observers can validate that the number of votes does not exceed the number of issued blind signatures. Finally, the Internal Observers can verify blind signatures from each voting transaction. However, the Internal Observers are not able to detect whether the adversary added votes on behalf of the Voters who did not actually vote.

Coercion-resistance is not satisfied. There are no mechanisms (e.g. re-voting) that protect from coercion. The Voter can, for example, record a video of the voting process such that the public key is also captured on this video. The adversary can later verify if the transaction containing this public key has been published into the Blockchain.

Receipt-freeness is satisfied. The Voter does not obtain a receipt that can be used to prove how they voted. The Voter's public key is not a receipt as it cannot be used to show how they voted.

Vote secrecy is satisfied if the Voting Device is not corrupted by the adversary. 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 [30].

Additionally, the Registrar and the Vote Collector must not collude. 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 learns 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 also compromise the Organiser and Decryptor (HSM module).

Fairness is satisfied if the Organiser and Tallier do not collude to restore the secret key before the voting phase has ended to decrypt intermediate results.

Dispute resolution is not satisfied as there are no procedures in place for the Voter to follow if they notice that the system is misbehaving. If the Voter notices that their voting transaction is missing from the Blockchain, they cannot re-vote and they can not prove that the system misbehaved. The Voter only has their public key which is not sufficient to prove that the Vote Collector or the Tallier misbehaved and dropped their voting transaction.

Additionally, we analysed the process of generating and using the final encryption/decryption key pair. The idea of creating shares of ElGamal secret key and performing distributed decryption is not novel, for example, it is used in the ElectionGuard system [31]. The usual approach is to select a set of trustees who independently generate their ElGamal key pairs. Next, the public keys are homomorphically combined to form a single ElGamal public key which is used to encrypt the votes. In this case, the secret key that can be used to decrypt the votes is never reconstructed from the shares. Trustees individually compute their partial decryptions, these partial decryptions are later combined to form full decryption of the election results.

However, the approach used in the Russian federal e-voting protocol is different. Firstly, the Organiser and the Decryptor independently generate their ElGamal key pairs. Next, the Organiser splits their existing secret key using Shamir Secret Sharing. Finally, the Organiser’s and the Decryptor’s public keys are combined to form a final encryption key as shown in (1). In the tallying phase, the Decryptor performs both partial decryptions using their secret key and the Organiser’s secret key. Unfortunately, this approach to the key generation was not thoroughly explained and justified in the documentation. It remains unclear, why the Organiser’s secret key share is generated and then split, instead of generating independent shares of the key from the beginning.

IV. 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 its initial analysis.

We found that the main weakness of the system lies in the authentication process. The usage of a password-based authentication system (ESIA) leads to a higher chance of different attacks. Furthermore, the attack surface increases as ESIA is used to authenticate users to more than 1000 IT systems [32].

Secondly, the analysis showed that the adversary will be able to break the vote secrecy of the system by compromising the Voting Device. Our analysis showed that the system does not provide individual verifiability. Therefore, if a corrupted Voting Device alters the vote, it will remain unnoticed by the Voter.

Additionally, the system uses a non-standard approach for generating the ElGamal encryption key. In the official documentation [12], it has not been specified if this approach is invented by the authors of the e-voting protocol or based on prior work. The better practice is to use more established and better understood cryptographic techniques such as distributed key generation for ElGamal presented in [31] 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.

V. Acknowledgements

This paper has been supported by the Estonian Research Council under the grant number PRG920 and European Regional Development Fund through the grant number EU48684. We would like to thank Kristjan Krips for his advice and expertise.

References

[1] D. L. Dill, B. Schneier, and B. Simons, “Voting and technology: who gets to count your vote?,” Commun. ACM, vol. 46, no. 8, pp. 29–31, 2003.

[2] J. Bannet, D. W. Price, A. Rudys, J. Singer, and D. S. Wallach, “Hack-a-vote: Security issues with electronic voting systems,” IEEE Security & Privacy, vol. 2, no. 1, pp. 32–37, 2004.

[3] A. J. Feldman, J. A. Halderman, and E. W. Felten, “Security analysis of the diebold accuvote-ts voting machine,” in Proceedings of the USENIX Workshop on Accurate Electronic Voting Technology, EVT’07, (USA), p. 2, USENIX Association, 2007.

[4] A. W. Appel, M. Ginsburg, H. Hursti, B. W. Kernighan, C. D. Richards, and G. Tan, “Insecurities and inaccuracies of the Sequoia AVC Advantage 9.00 H DRE voting machine,” 2008.

[5] A. Aviv, P. Černy, S. Clark, E. Cronin, G. Shah, M. Sherr, and M. Blaze, “Security evaluation of es’&amp;s voting machines and election management system,” in Proceedings of the Conference on Electronic Voting Technology, EVT’08, (USA), USENIX Association, 2008.

[6] A. W. Appel, M. Ginsburg, H. Hursti, B. W. Kernighan, C. D. Richards, G. Tan, and P. Venetis, “The new jersey voting-machine lawsuit and the avc advantage dre voting machine,” in Proceedings of the 2009 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections, EVT/WOTE’09, (USA), p. 5, USENIX Association, 2009.

[7] C. Vegas and J. Barrat, "Overview of current state of e-voting worldwide," in Real-World Electronic Voting, pp. 67–92, 6000 Broken Sound Parkway, NW, (Suite 300): Auerbach Publications, 2016.

[8] J. P. Gibson, R. Krimmer, V. Teague, and J. Pomares, "A review of E-voting: the past, present and future," Ann. des Télécommunications, vol. 71, no. 7-8, pp. 279–286, 2016.

[9] P. Gaudry and A. Golovnev, "Breaking the encryption scheme of the moscow internet voting system," in Financial Cryptography and Data Security (J. Bonneau and N. Heninger, eds.), (Cham), pp. 32–49, Springer International Publishing, 2020.

[10] Департамент города Москвы по конкурентной политике, "Протокол подведения итогов открытого конкурса в электронной форме от 12.05.2021 №ППИ1," May 2021.

[11] Mikhail Tetkin, RBC, "Ростелеком» разработает систему голосования на блокчейне по заказу ЦИК," August 2020.

[12] Central Election Commission of Russia, "Дистанционное электронное голосование," 2021.

[13] Consult Partner, "Making snils to a foreign citizen," 2021.

[14] Ministry of Digital Development, Communications and Mass Media of the Russian Federation, "Единая система идентификации и аутентификации (ЕСИА)," 2021.

[15] Waves Enterprise, "Семинар «Технологии блокчейн и криптозащиты в системе ДЭГ," 2021.

[16] Waves Enterprise, "Technical description of the waves enterprise voting," 2021.

[17] Central Election Commission of Russia, "Описание протокола ДЭГ к выборам, голосование на которых состоится 17, 18 и 19 сентября 2021 г.," 2021.

[18] Central Election Commission of Russia, "Описание ПТК ДЭГ," 2021.

[19] Central Election Commission of Russia, "Порядок дистанционного электронного голосования на выборах, назначенных на 19 сентября 2021 года," 2021.

[20] A. D. V. Dolmatov, "Gost r 34.11-2012: Hash function," August 2013.

[21] M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko, "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.

[22] T. El Gamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in Proceedings of CRYPTO 84 on Advances in Cryptology, (Berlin, Heidelberg), p. 10–18, Springer-Verlag, 1985.

[23] A. Shamir, "How to share a secret," Commun. ACM, vol. 22, p. 612–613, November 1979.

[24] A. D. V. Dolmatov, "Gost r 34.10-2012: Digital signature algorithm," December 2013.

[25] D. Chaum and T. P. Pedersen, "Wallet databases with observers," in Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO '92, (Berlin, Heidelberg), p. 89–105, Springer-Verlag, 1992.

[26] S. V. Smyshlyaev, E. Alekseev, I. Oshkin, V. Popov, S. Leontiev, V. Podobaev, and D. Belyavsky, "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.

[27] A. Juvonen, "A framework for comparing the security of voting schemes," 2019.

[28] K. Krips and J. Willemson, "On practical aspects of coercion-resistant remote voting systems," in Electronic Voting (R. Krimmer, M. Volkamer, V. Cortier, B. Beckert, R. Küsters, U. Serdült, and D. Duenas-Cid, eds.), (Cham), pp. 216–232, Springer International Publishing, 2019.

[29] S. Neumann, Evaluation and Improvement of Internet Voting Schemes Based on Legally-Founded Security Requirements. PhD thesis, Technische Universität Darmstadt, 03 2016.

[30] S. Heiberg, K. Krips, and J. Willemson, "Mobile voting – still too risky?," in Financial Cryptography and Data Security. FC 2021 International Workshops (M. Bernhard, A. Bracciali, L. Gudgeon, T. Haines, A. Klages-Mundt, S. Matsuo, D. Perez, M. Sala, and S. Werner, eds.), (Berlin, Heidelberg), pp. 263–278, Springer Berlin Heidelberg, 2021.

[31] M. N. Josh Benaloh, "Electionguard specification v1.0," 2022.

[32] Identity Blitz, "Russian e-government system of trusted identities," 2021.