CYBERNETICA

Analysis of Russian Federal Remote E-voting Scheme of 2021

Jelizaveta Vakarjuk, **Nikita Snetkov**, Jan Willemson

Cybernetica AS

May 9, 2022

Agenda

- Small intro
- Description of system
- Analysis
- Conclusions

Gosduma 2021 – two systems

- Moscow had a separate system that was based on 2019 and 2020 schems (about this one you probably heard from the news). Developed by DIT and Kaspersky.

Gosduma 2021 – two systems

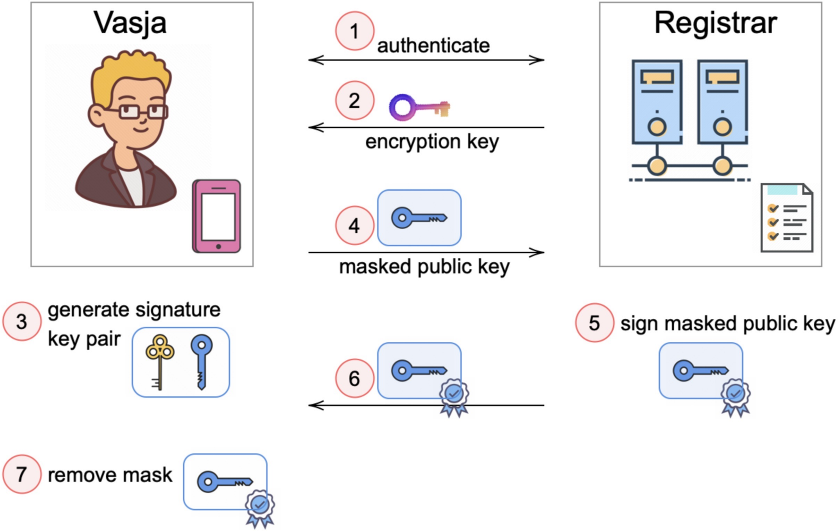
- Moscow had a separate system that was based on 2019 and 2020 schems (about this one you probably heard from the news). Developed by DIT and Kaspersky.
- Kursk region, Murmansk region, Nizhny Novgorod region, Rostov region, Yaroslavl region and Sevastopol had a system developed by Waves Enterprise and Rostelecom. (This is the one we were researching.)

Part 1. Description

Main participants

- Voter
- Organiser
- Registrar
- Vote Collector
- Tallier = Blockchain + Decryptor
- (Election Observer)

Authorisation phase



Vasja

Registrar

1 authenticate

2 encryption key

3 generate signature key pair

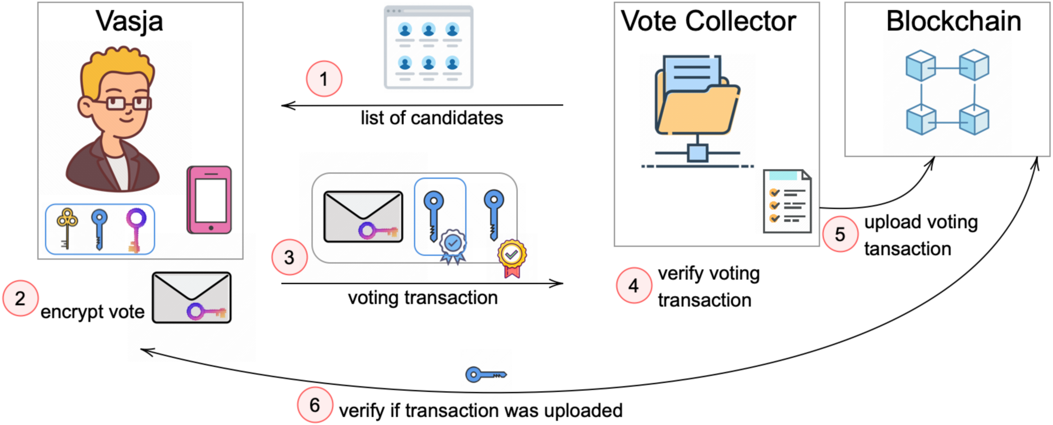
4 masked public key

5 sign masked public key

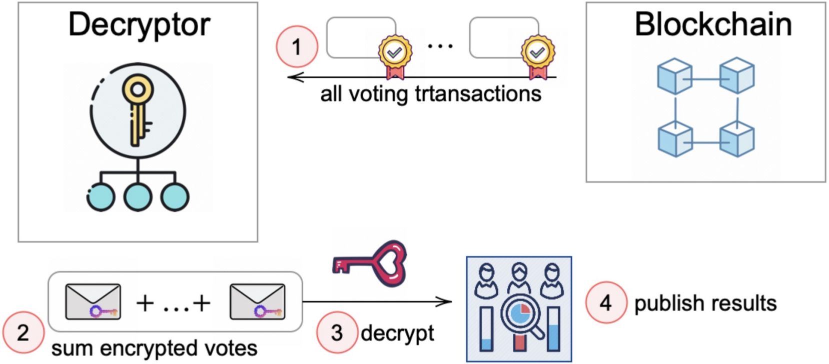
6

7 remove mask

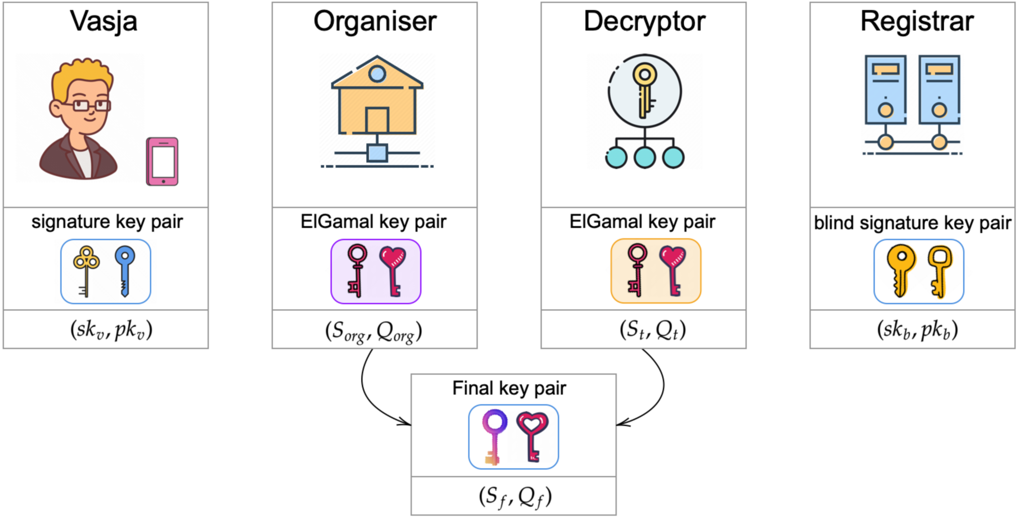
Voting phase



Tallying phase



Why so many keys?



Vasja

signature key pair

(sk_v, pk_v)

Organiser

ElGamal key pair

(S_{org}, Q_{org})

Decryptor

ElGamal key pair

(S_t, Q_t)

Registrar

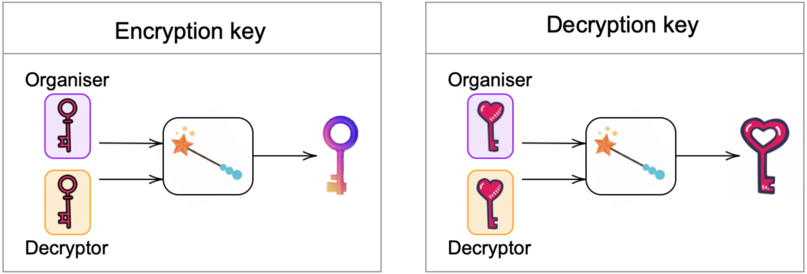
blind signature key pair

(sk_b, pk_b)

Final key pair

(S_f, Q_f)

One caveat...



Encryption key

Organiser

Decryptor

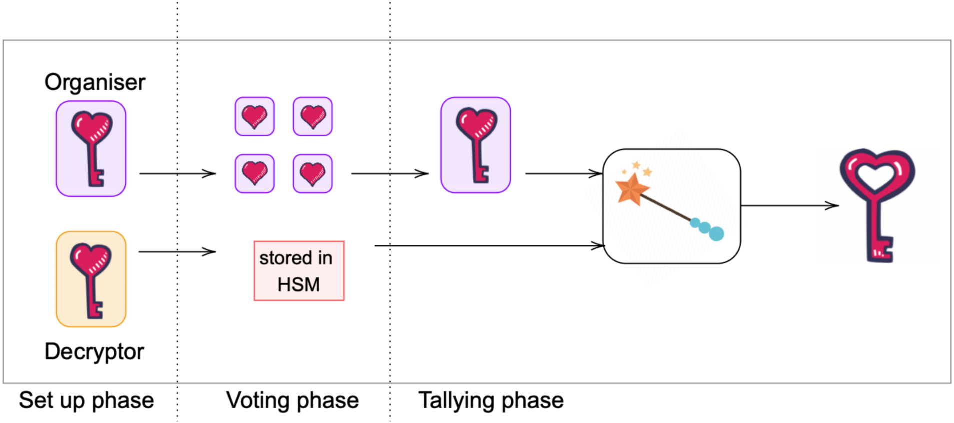
Decryption key

Organiser

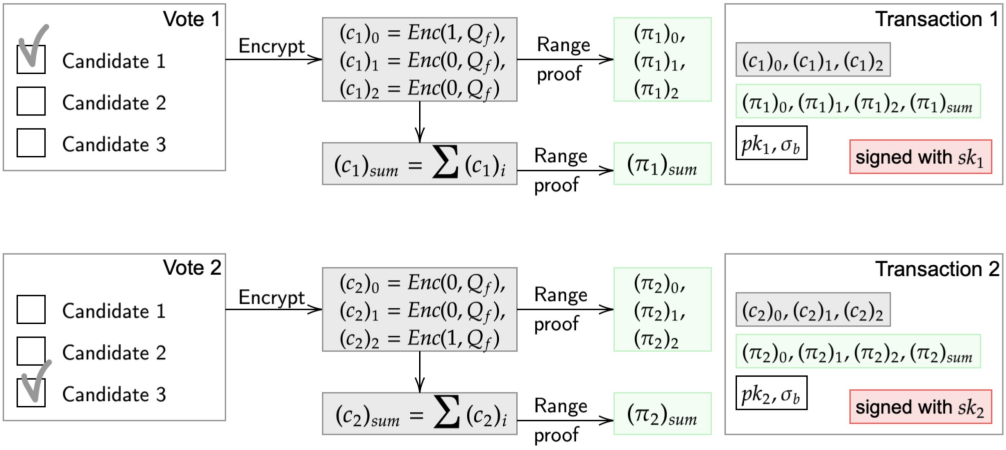
Decryptor

We do not want any party to hold decryption key 🔑 before the tallying phase

One caveat...


Organiser
stored in HSM
Decryptor
Set up phase
Voting phase
Tallying phase

Encrypting vote



Vote 1

☑ Candidate 1

□ Candidate 2

□ Candidate 3

Encrypt

$(c_1)_0 = Enc(1, Q_f),$
$(c_1)_1 = Enc(0, Q_f),$
$(c_1)_2 = Enc(0, Q_f)$

Range proof

$(\pi_1)_0,$
$(\pi_1)_1,$
$(\pi_1)_2$

$(c_1)_{sum} = \sum (c_1)_i$

Range proof

$(\pi_1)_{sum}$

Transaction 1

$(c_1)_0, (c_1)_1, (c_1)_2$

$(\pi_1)_0, (\pi_1)_1, (\pi_1)_2, (\pi_1)_{sum}$

pk_1, σ_b

signed with sk_1

Vote 2

□ Candidate 1

□ Candidate 2

☑ Candidate 3

Encrypt

$(c_2)_0 = Enc(0, Q_f),$
$(c_2)_1 = Enc(0, Q_f),$
$(c_2)_2 = Enc(1, Q_f)$

Range proof

$(\pi_2)_0,$
$(\pi_2)_1,$
$(\pi_2)_2$

$(c_2)_{sum} = \sum (c_2)_i$

Range proof

$(\pi_2)_{sum}$

Transaction 2

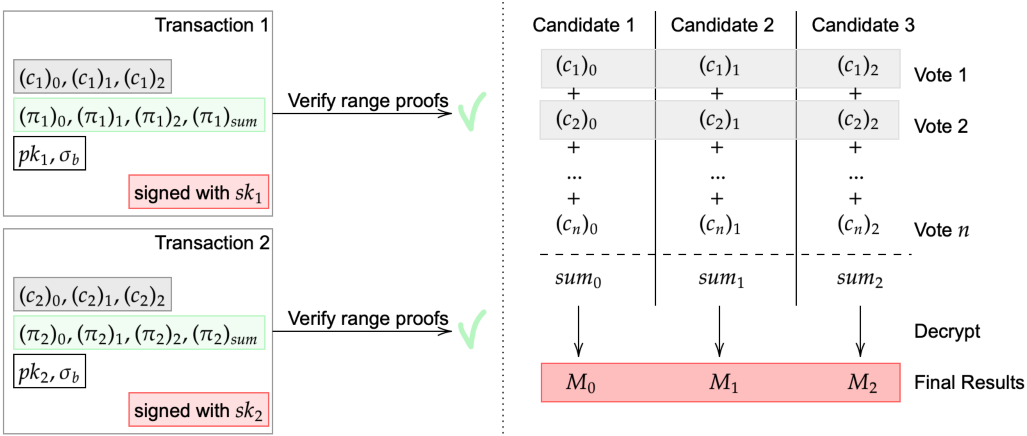
$(c_2)_0, (c_2)_1, (c_2)_2$

$(\pi_2)_0, (\pi_2)_1, (\pi_2)_2, (\pi_2)_{sum}$

pk_2, σ_b

signed with sk_2

Tallying phase



Transaction 1

$(c_1)_0, (c_1)_1, (c_1)_2$

$(\pi_1)_0, (\pi_1)_1, (\pi_1)_2, (\pi_1)_{sum}$

pk_1, σ_b

signed with sk_1

Verify range proofs ✓

Transaction 2

$(c_2)_0, (c_2)_1, (c_2)_2$

$(\pi_2)_0, (\pi_2)_1, (\pi_2)_2, (\pi_2)_{sum}$

pk_2, σ_b

signed with sk_2

Verify range proofs ✓

Candidate 1	Candidate 2	Candidate 3	
$(c_1)_0$	$(c_1)_1$	$(c_1)_2$	Vote 1
+	+	+	
$(c_2)_0$	$(c_2)_1$	$(c_2)_2$	Vote 2
+	+	+	
...	...	...	
+	+	+	
$(c_n)_0$	$(c_n)_1$	$(c_n)_2$	Vote n
sum_0	sum_1	sum_2	
↓	↓	↓	Decrypt
M_0	M_1	M_2	Final Results

Part 2. Analysis

Individual verifiability

The Voter should be able to verify that their vote was correctly interpreted and successfully recorded without alterations.

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.

Universal verifiability

Everyone should be able to verify that the final tally is correctly calculated from the published votes.

Satisfied Election Observers can verify the correctness of tally using public information from the Blockchain.

Eligibility

The system should ensure that only eligible voters are allowed to cast a vote.

Partially satisfied There is no strong cryptographic identification in place in Russia. The whole authentication process relies on the gosuslugi.ru identification service, and in the weakest instances, just one password is enough to access it.

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.

Not satisfied There are no mechanisms (e.g. re-voting) that would protect the Voter from over-the-shoulder coercion.

Receipt-freeness

The Voter should not be able to produce proof to a coercer that they voted in a particular manner.

Satisfied The protocol itself does not produce a receipt. However, there exists a side channel of the Voter recording her voting session (but this is not considered a receipt in the cryptographic protocol sense).

Vote secrecy

It should be impossible to link the content of the cast vote to the Voter’s identity.

Partially satisfied if the Voting Device is not corrupt and the Voter herself does not breach vote secrecy on purpose. Additionally, the Registrar and the Vote Collector must not collude.

Fairness

It should not be possible to calculate intermediate results of an election before the tallying phase has ended.

Satisfied as the all the private key components are needed for tally.

Dispute resolution

If the Voter notices malicious behavior of the voting system, they should be able to prove it.

Not satisfied 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.

Conclusions

- The protocol has been composed by someone who obviously knows about cryptography and voting protocols.
- Voter authentication is the weakest point.
- There are several security assumptions (e.g. the Voting Device should not be corrupted).
- A non-standard approach for generating the ElGamal encryption key has been selected for some reason.

Questions?