

# zkVoting : Zero-knowledge proof based coercion-resistant and E2E verifiable e-voting system and its application to National Election Commission of South Korea

**Seongho Park**

Hanyang University



**HANYANG UNIVERSITY**

**Jaekyoung Choi**

Zkrypto Inc.



KOOKMIN UNIVERSITY

**Jihye Kim**

Kookmin University &  
Zkrypto Inc.

**Hyunok Oh**

Hanyang University &  
Zkrypto Inc.



**Paper**

<https://ia.cr/2024/1003>



**Web**

<https://zkvoting.com>

ZKP Standard, March 24, 2025

# Contents

---

1. Motivation & Adoption Journey
2. Technical Deep Dive

# Motivation

## ❖ Key technology to solve global challenges


**CES** Consumer Technology Association  
JAN 5-8, 2023  
LAS VEGAS, NV & DIGITAL  
CES 2023 INNOVATION AWARD PRODUCT

TOPICS ABOUT SESSIONS & EVENTS EXHIBITS LOGISTICS

**REGISTER**

### zkVoting


By Zkrypto Inc. / Hanyang University



**Best of Innovation**  
Cybersecurity & Personal Privacy

**Honoree**  
Software & Mobile Apps

zkVoting is the first public blockchain voting app to utilize ZKP (Zero Knowledge Proof) technology. We leverage the trusted public blockchain ecosystem and ZKP technology to deliver the first voting platform that ensures absolute secrecy, guaranteed legitimate information without revealing voter identification and ballot content, and is resistant to adversarial coercion. With our intuitive and user-friendly zkVoting mobile app and accurate validation of ballot results, we are unleashing a new era for legitimate elections.



The world's most SECURE & TRANSPARENT  
**zkVoting**



**CES** 2024 INNOVATION AWARD PRODUCT

### zkVoting: Blockchain-based voting at the Poll Station

By Zkrypto Inc.



**Best of Innovation**  
Cybersecurity & Personal Privacy

zkVoting poll station is the first in-person blockchain-powered voting system. Recognised with the Best of Innovation award at CES 2023, zkVoting introduces seamless on-site voting support. Voters can now verify the accurate casting of their ballots in real-time at the poll station. Our zero-knowledge proof protocol enables anyone to verify vote validity while upholding voter privacy. We separate the voting from vote validation devices and utilise a "fake key" generation to prevent malware and foster process credibility. zkVoting empowers citizens to engage confidently in online or in-person voting, ensuring transparency, security, and advancing democracy.



Blockchain-based Voting System with a Supported Poll Station  
**zkVoting**



<https://www.youtube.com/watch?v=askp6ZIGpvs&t=2333s>

# Motivation

## ❖ Why CES focuses on voting systems

Rising doubt on voting systems



Protesters break into U.S. Capitol in January 2021



Supporters of former President Jair Bolsonaro clash with security forces as they raid the National Congress in Brasília

Protesters break into Brazilian Capitol and presidential palace in January 2023



# Motivation

## ❖ Why CES focuses on voting systems

Rising doubt on voting systems



In December 2024, South Korean president declared emergency martial law, claiming **election fraud** in the parliamentary elections.



Martial forces seized the National Assembly and the **National Election Commission**, where they accessed servers and detained NEC staff.

Conspiracy theories about hacking and manipulation continue to spread across social media, fueling distrust and polarization.

**Can we fully secure and verify our elections?**

# Motivation

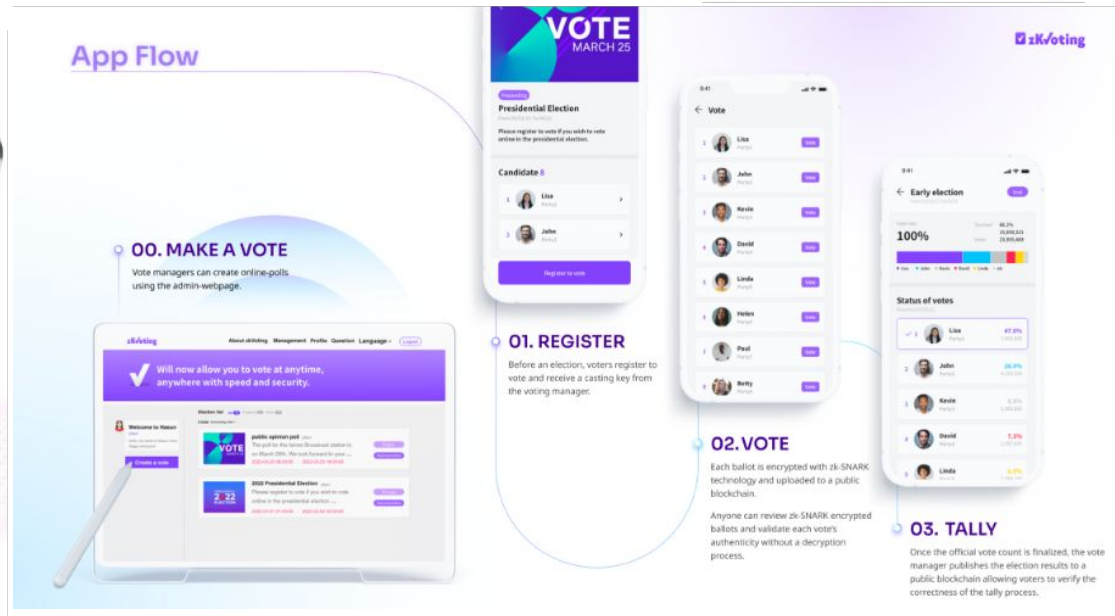
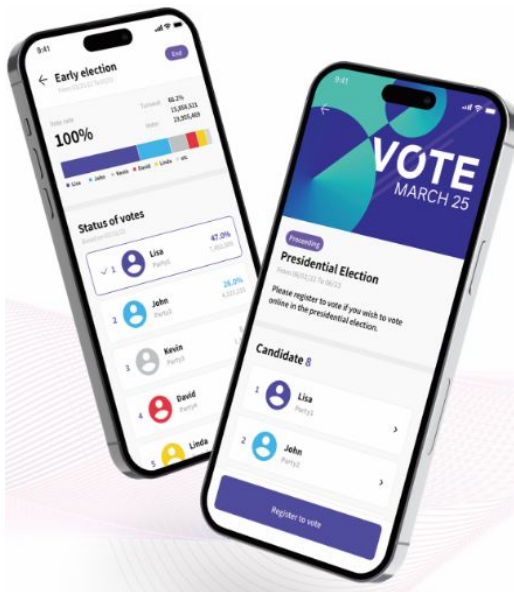
## ❖ zkVoting: Securing Elections with Zero-Knowledge Proofs and Blockchain

- Provides **coercion-resistant** secret voting and **end-to-end verifiability** for voters, and prevents vote manipulation based on public blockchain and zero-knowledge proof
- Officially adopted by the **National Election Commission** of Korea for public sector elections



| Cybersecurity & Personal Privacy |

| Software & Mobile Apps |



## ❖ History of how we have started this project

- **2018:** NEC began integrating blockchain technology into its online voting system
- **2022:** Mandatory use of Zero-Knowledge Proofs (ZKP)
  - NEC implemented a blind signature-based approach
  - Limitations: Could not verify encrypted ballots or the final tally
- **Challenge:** Difficult to introduce programmable ZKP voting systems
- **2023:** CES Best of Innovation Award enables NEC collaboration
  - Presented benefits of programmable ZKPs:
    - End-to-end verification (registration → tally)
    - Secure and auditable system design
- **November 2024:** Launch of new ZKP-based blockchain voting system with NEC

## ❖ Where zkVoting is Used

✓ **Currently deployed for public**

### **Sector Elections:**

- Public institution leader elections
- Local community referenda
- Political party candidate nominations

✗ **Not yet applied to:**

- Presidential elections
- Parliamentary elections
- Local government elections

## ❖ Current Status

- Service temporarily on hold due to snap presidential election preparation after impeachment.

## ❖ Major Upcoming Elections (2025)

- People Power Party Leader Election (~800,000 voters)
- Jeju Island Referendum (~560,000 voters)

## Voting Statistics (Dec 2024 – Jan 2025)

• Over **100,000 total votes** cast since public sector launch (Nov 30, 2024)

• **Daily Peak Participation:** 15,043 voters on Dec 27

• **Monthly Breakdown:**

Month	Elections	Voters	Participants
Dec 2024	220	102,208	84,338
Jan 2025	47	20,901	16,827



### ❖ Challenges in zkVoting approval in Korea

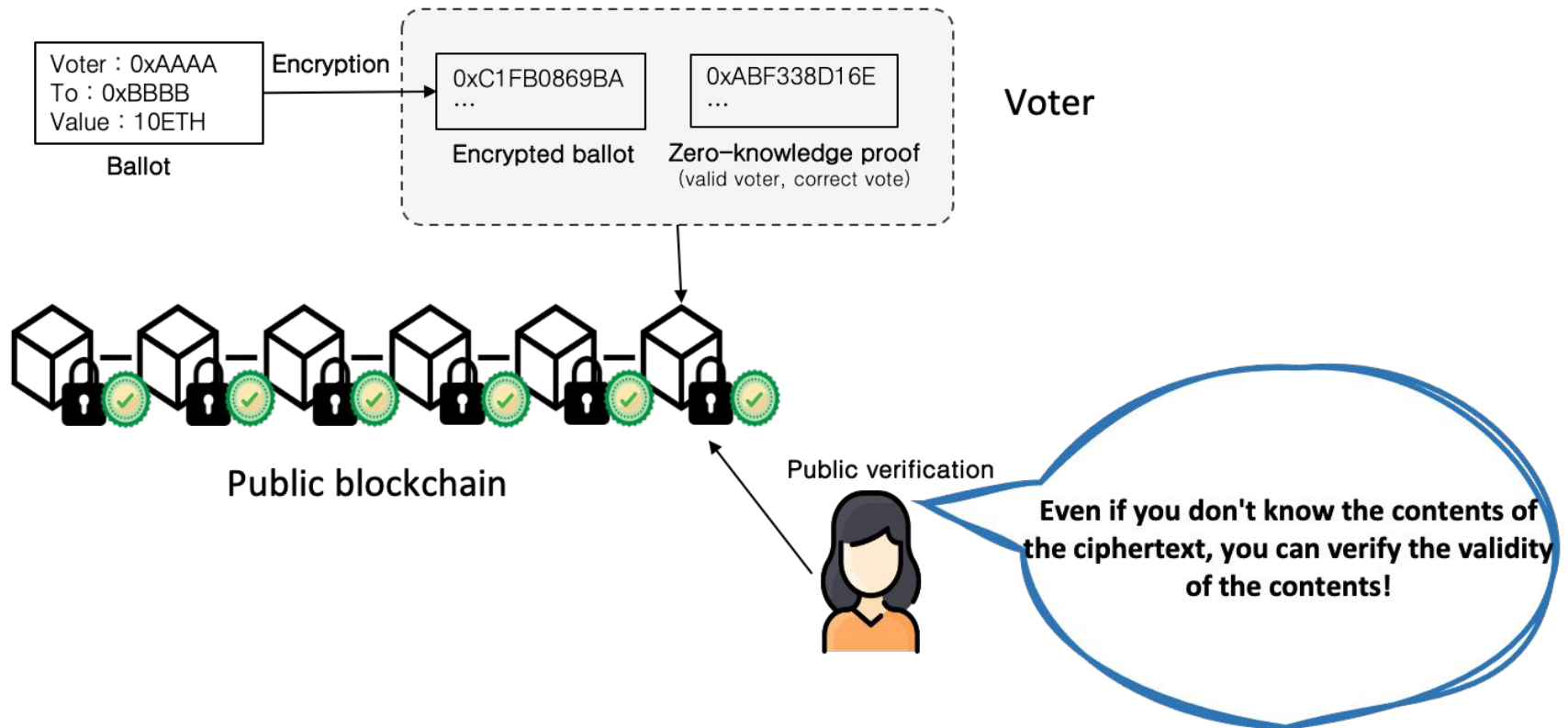
- **Lack of standards for ZKP and other cryptographic primitives** caused difficulties in the National Election Commission approval
- Current approval process relies on:
  - Standard cryptographic algorithm certifications.
  - Extensive source code security evaluations.

### ❖ What we need to do

- Define **standards for the cryptographic primitives** in ZKP
  - ZKP friendly hash functions : MiMC, Poseidon hash, ...
  - ZKP protocols : Groth16, Plonk, KZG, ...
- Establish a **ISO certification framework** for ZKP

# Technical Overview

- Storing ballots on a public blockchain for public verification
- Since the ballot on the public blockchain is publicly available, it is no secret voting
- If the ballot is encrypted in the public blockchain, its privacy is preserved but its correctness is not guaranteed
- Zero-knowledge proof guarantees voter validity and ballot correctness without the content of the ballot



# Technical Overview

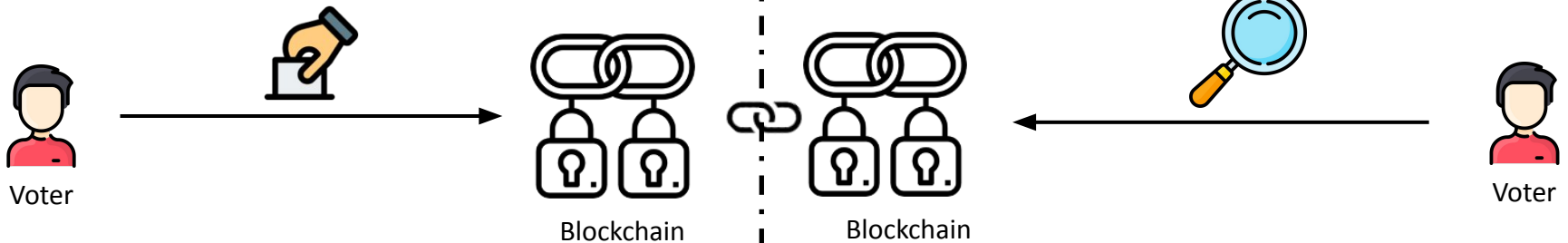
## ❖ Goal



### Privacy



### End-to-End verifiability



Voter can hide the message in the ballot, evading coercion attack.  
The vote transaction can be anonymous

Voter can verify that their ballot is cast-as-intended, recorded-as-cast, and tallied-as-recorded.



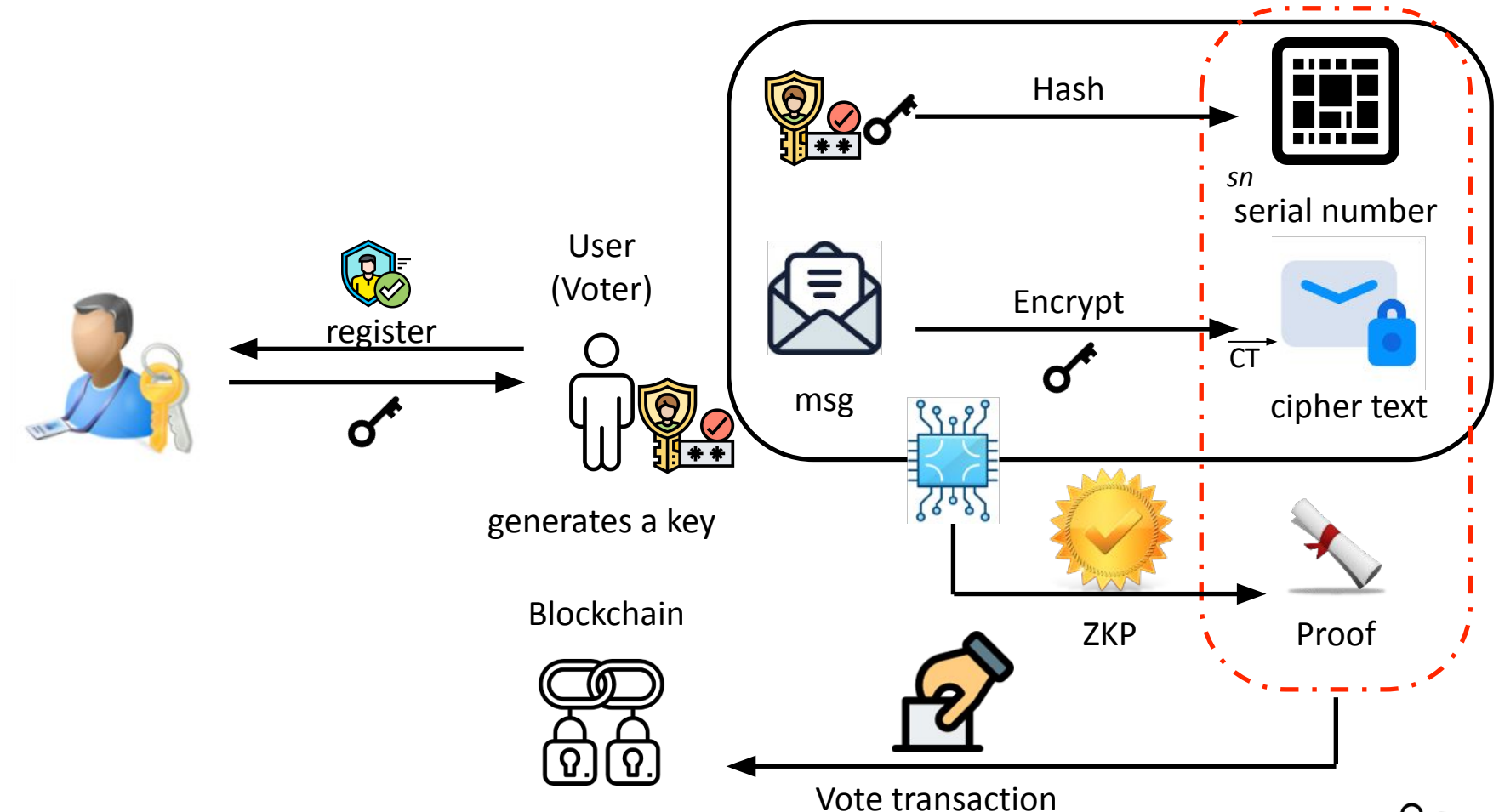
### Efficiency

- Succinct time to cast a ballot (2.3s in iPhone14 Pro)
- Time complexity for the tally phase is linear to the number of ballots :  $O(n_b)$

# Approach

## ❖ Our approach

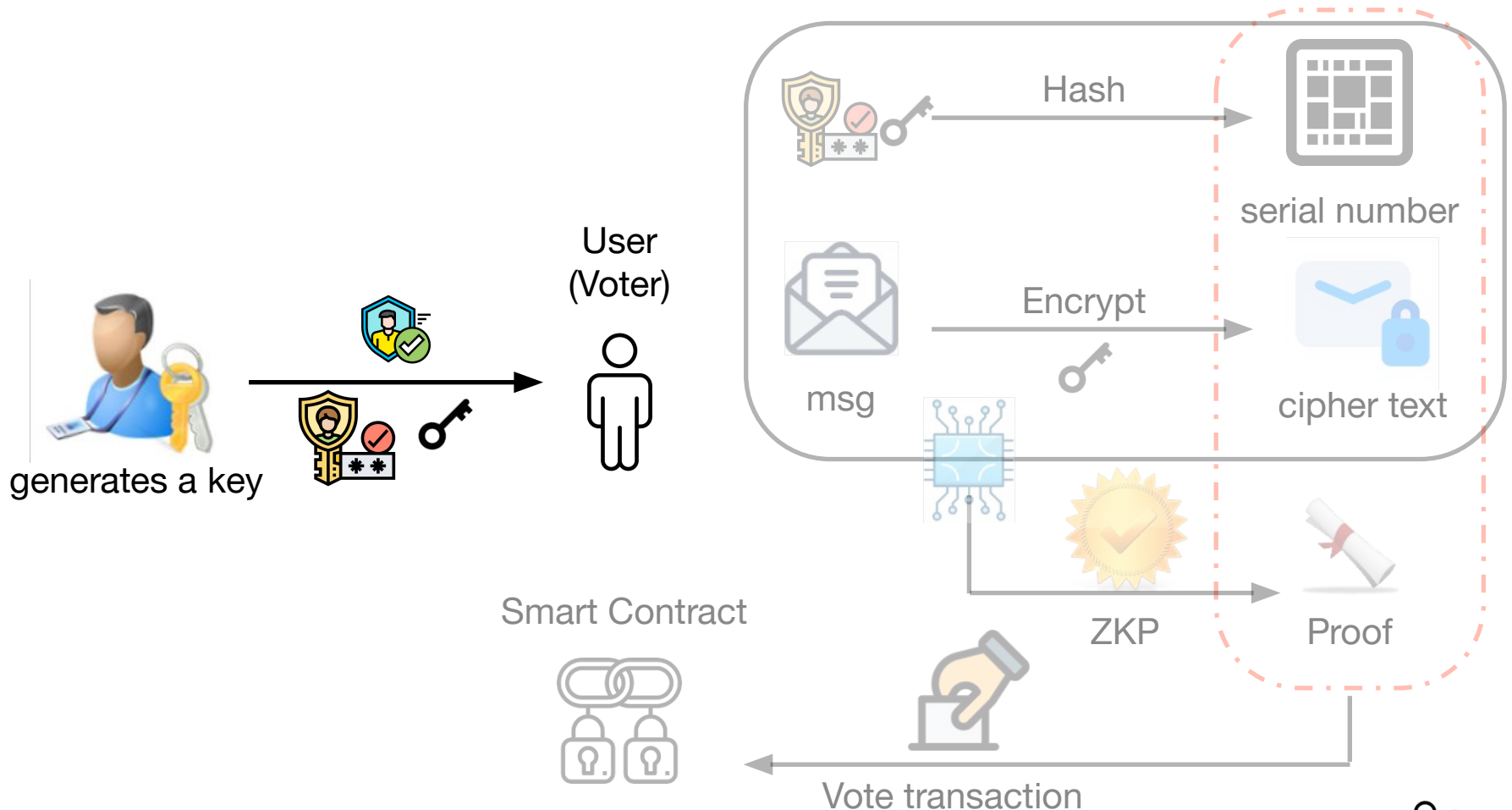
- Registration step : Voters generate their own keys, register the public key to the election authority and acquire a casting key
- Casting step : Voters compute sn, and CT with ZKP



# Requirement

## ❖ Our approach

- Voters may struggle with securely generating and storing keys
- NEC's requirement : it generates all keys and issues them to voters

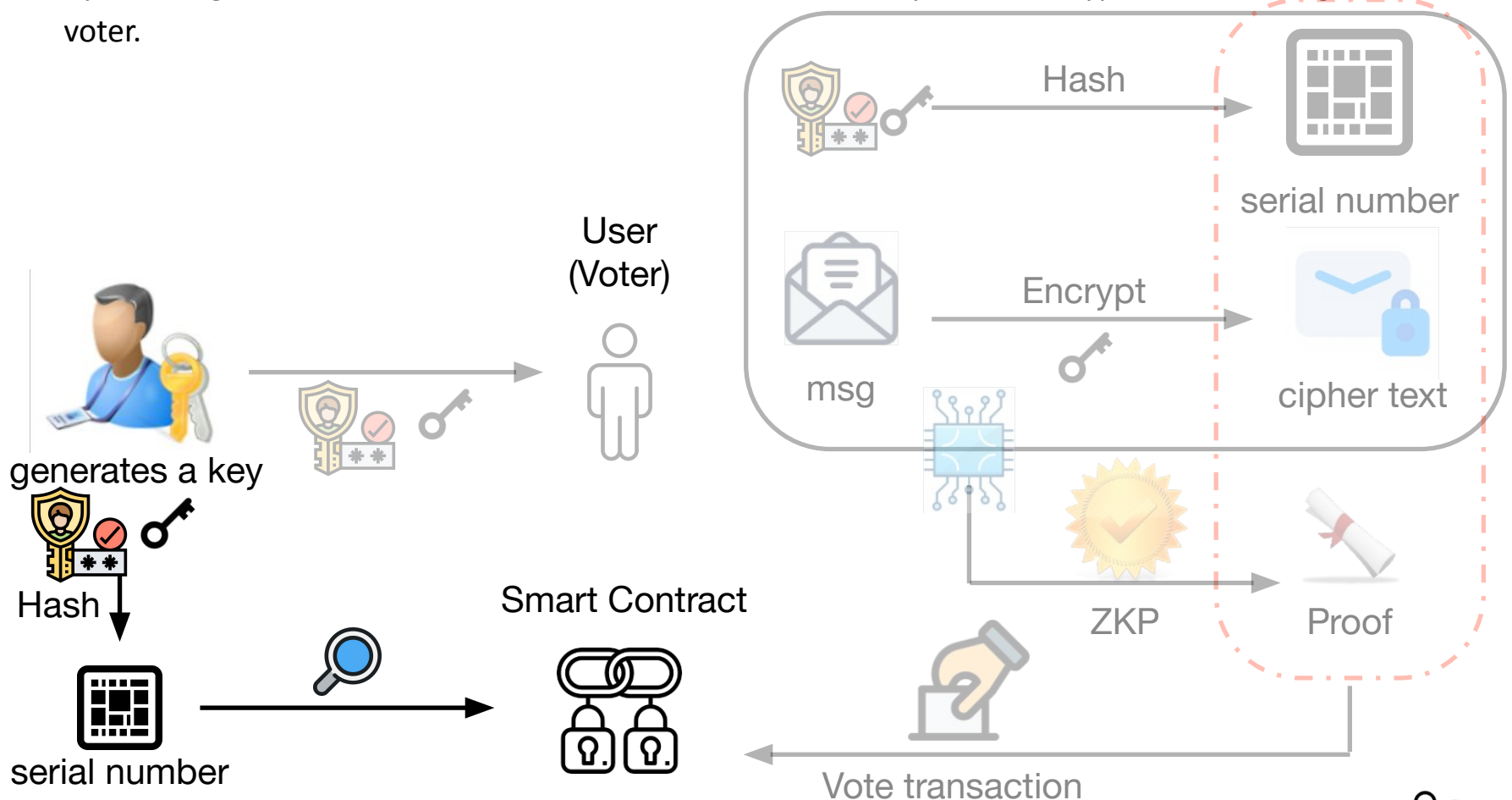




# Privacy Problem

## ❖ Our approach

- NEC generates both public and private keys for voters.
- This breaks secret voting guarantees.
- NEC can link serial numbers to voters using private keys stored on their servers.
- By checking serial numbers on the blockchain, NEC can identify which encrypted vote belongs to which voter.



## ❖ Our approach

- Randomize the vote transaction in two parts to break the linkability and ensure secret voting



Vote transaction

$$tx = (sn, \overrightarrow{CT}, \pi)$$

## ❖ Our approach

- Randomize the vote transaction in two parts



Vote transaction

$$tx = (sn, \overrightarrow{CT}, \pi)$$

randomize sn  $sn' = rand(sn)$

$$tx_1 = (sn', \overrightarrow{CT}, \pi)$$

- The authority, by looking at alone, **cannot determine the owner** of the transaction

## ❖ Our approach

- Randomize the vote transaction in two parts



Vote transaction

$$tx = (sn, \overrightarrow{CT}, \pi)$$

randomize sn  $sn' = rand(sn)$

$$tx_1 = (sn', \overrightarrow{CT}, \pi)$$

randomize CT  $CT' = rand(CT)$

$$tx_2 = (sn, CT', \pi)$$

- The authority, by looking at alone, **cannot determine the message** of the transaction

## ❖ Our approach

- Randomize the vote transaction in two parts



Vote transaction

randomize sn  $sn' = rand(sn)$

$$tx_1 = (sn', \overrightarrow{CT}, \pi)$$

$$tx = (sn, \overrightarrow{CT}, \pi)$$

randomize CT  $CT' = rand(CT)$

$$tx_2 = (sn, CT', \pi)$$

- The voter can **preserve privacy** unless the **authority recognizes that two transactions originate from the same source transaction**



## ❖ Our approach

- Randomize the vote transaction in two parts



Vote transaction

randomize sn  $sn' = rand(sn)$

$$tx_1 = (sn', \overrightarrow{CT}, \pi)$$

$$tx = (sn, \overrightarrow{CT}, \pi)$$

randomize CT  $CT' = rand(CT)$

$$tx_2 = (sn, CT', \pi)$$

- The input of each transaction has been changed.
- Each transaction would not be verified

## ❖ Our approach

- Randomize the vote transaction in two parts



Vote transaction

$$tx = (sn, \overrightarrow{CT}, \pi)$$

randomize sn  $sn' = rand(sn)$

$$tx_1 = (sn', \overrightarrow{CT}, \pi)$$

randomize CT  $CT' = rand(CT)$

$$tx_2 = (sn, CT', \pi)$$

- The input of each transaction has been changed.
- Each transaction would not be verified



- Adjust proofs so that the transaction can be verified with randomized inputs

## ❖ Our approach



Vote transaction

$$tx = (sn, \overrightarrow{CT}, \pi)$$

randomize sn  $sn' = rand(sn)$

randomize CT  $CT' = rand(CT)$

$$tx_1 = (sn', \overrightarrow{CT}, \pi)$$

$$tx_2 = (sn, CT', \pi)$$

Adjust each proof to be verified with randomized inputs

$$tx_1 = (sn', \overrightarrow{CT}, \pi_1)$$

$$tx_2 = (sn, CT', \pi_2)$$

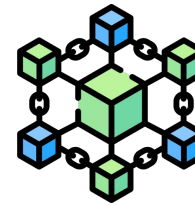
Each randomized transaction can be verified by itself!

## ❖ Our approach

$$tx_1 = (sn', \overrightarrow{CT}, \pi_1)$$



$$tx_2 = (sn, CT', \pi_2)$$



- Upload randomized transactions to the blockchain and the database separately to **prevent authorities from recognizing that two transactions originate from the same source transaction.**

## ❖ Our approach



Vote transaction

$$tx = (sn, \overrightarrow{CT}, \pi)$$

randomize sn  $sn' = rand(sn)$

randomize CT  $CT' = rand(CT)$

$$tx_1 = (sn', \overrightarrow{CT}, \pi)$$

$$tx_2 = (sn, CT', \pi)$$

Adjust each proof to be verified with randomized inputs

$$tx_1 = (sn', \overrightarrow{CT}, \pi_1)$$

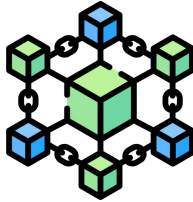
$\pi_c$

$$tx_2 = (sn, CT', \pi_2)$$

An connection proof  $\pi_c$  is generated with ensuring that  $tx_1$  and  $tx_2$  are originated from a single  $tx$ .



## ❖ Our approach



$$\begin{aligned} tx_1 &= (sn', \overrightarrow{CT}, \pi_1) \\ tx_1 &= (sn', CT, \pi_1) \\ tx_1 &= (sn', CT, \pi_1) \\ tx_1 &= (sn', CT, \pi_1) \\ &\vdots \end{aligned}$$



$$\begin{aligned} tx_2 &= (sn, CT', \pi_2) \\ tx_2 &= (sn, CT', \pi_2) \\ tx_2 &= (sn, CT', \pi_2) \\ tx_2 &= (sn, CT', \pi_2) \\ &\vdots \end{aligned}$$

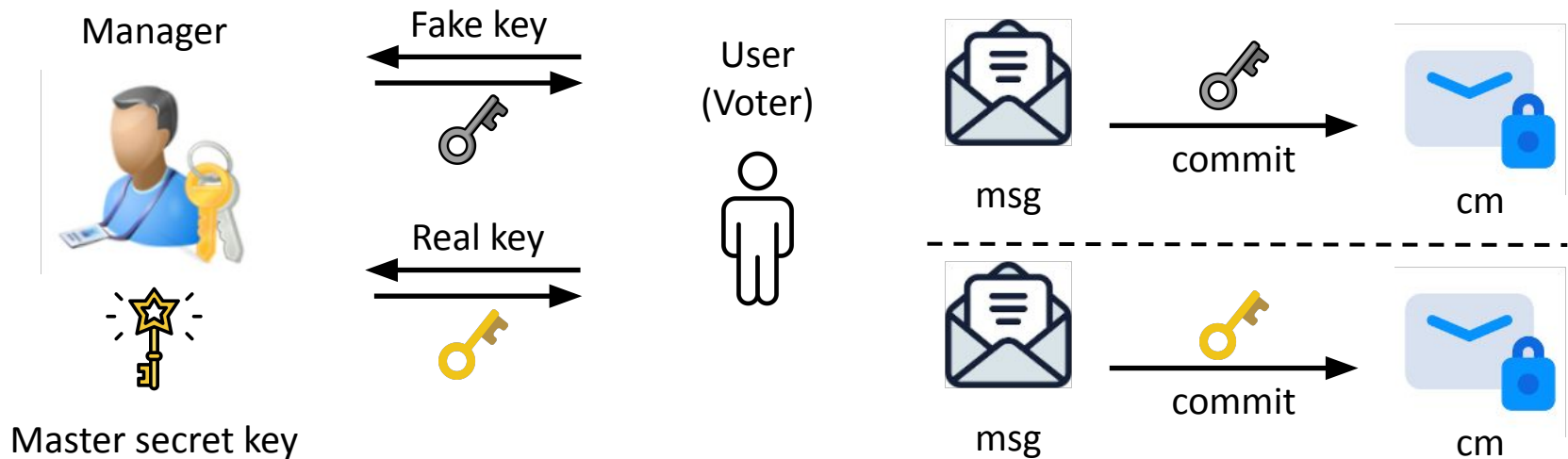
- Aggregate the connection proofs to a single aggregated connection proof
- Verify that **two transaction sets are the same set** using the aggregated connection proof

# Coercion Resistance

## ❖ Coercion resistance

- voters resist when a coercer forces them to vote for a specific candidate.
- Fake key vs. Revote

## ❖ Fake key : Nullifiable commitment scheme



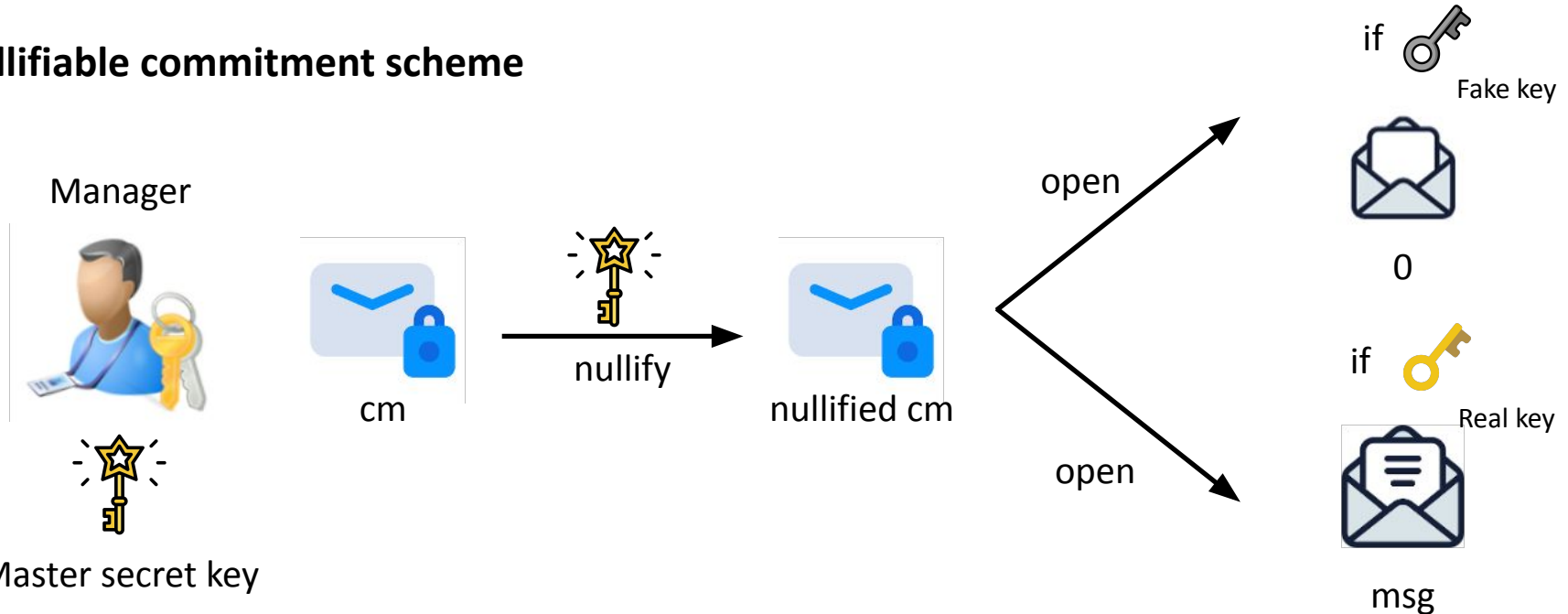
- All keys are indistinguishable
- A commitment cm satisfies hiding and binding

# Coercion Resistance

## ❖ Our work

- A new cryptographic primitive, **nullifiable commitment scheme**

## ❖ Nullifiable commitment scheme

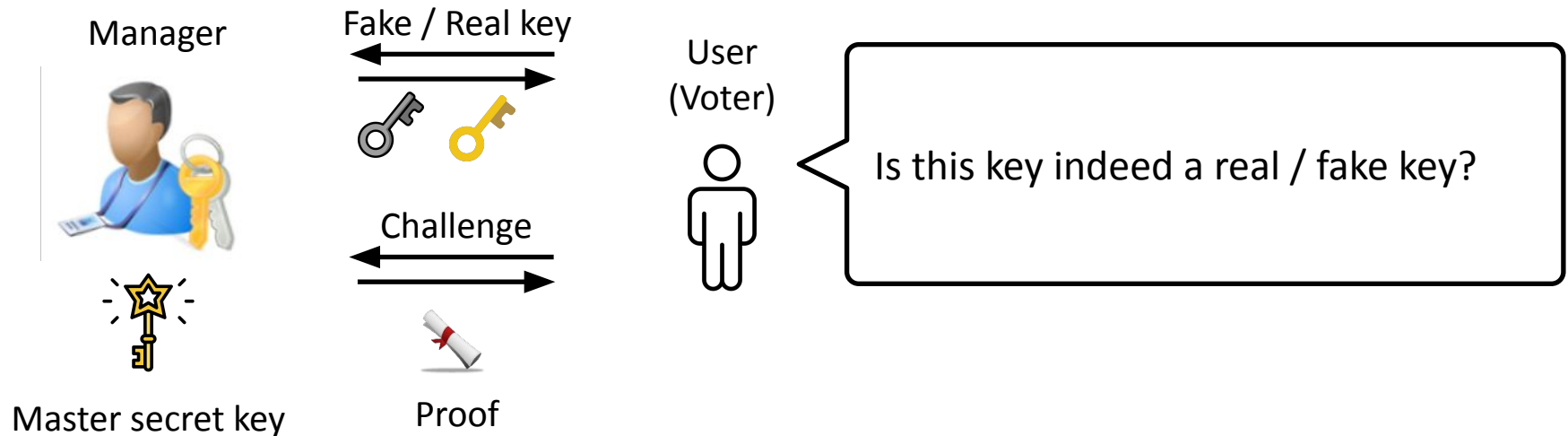


- A manager who holds master secret key can nullify a commitment
- A nullified commitment is opened as a zero if the original cm was generated with a fake commitment key. Otherwise, it will be opened as an original message.

## ❖ Our work

- A new cryptographic primitive, **nullifiable commitment scheme**

## ❖ Nullifiable commitment scheme



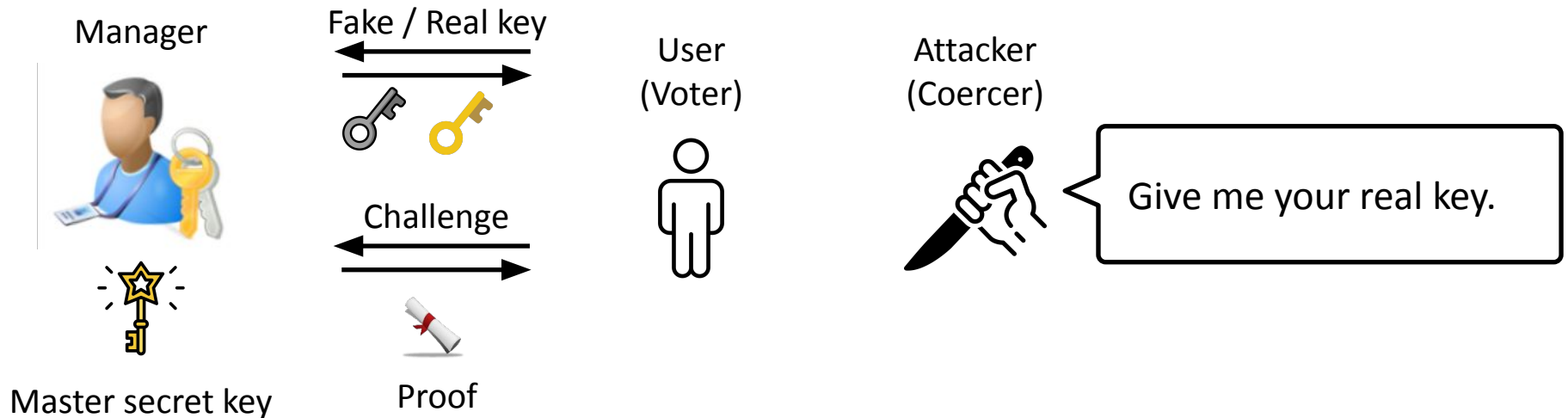
- A manager who holds master secret key can interactively prove to a casting key holder whether the key is real or fake.

# Coercion Resistance

## ❖ Our work

- A new cryptographic primitive, **nullifiable commitment scheme**

## ❖ Nullifiable commitment scheme



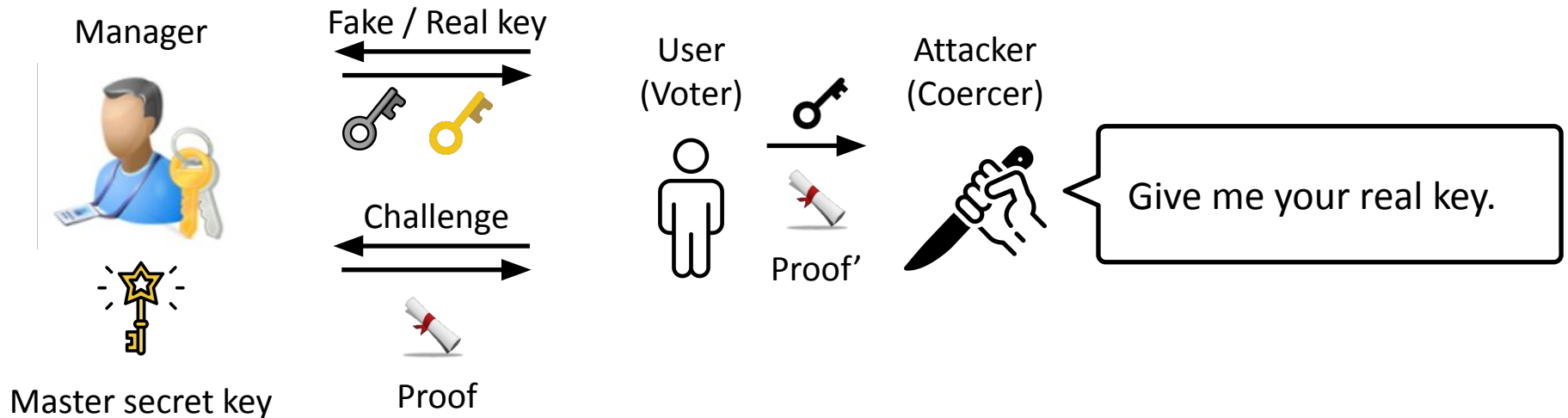


# Coercion Resistance

## ❖ Our work

- A new cryptographic primitive, **nullifiable commitment scheme**

## ❖ Nullifiable commitment scheme



- Any commitment key holder can simulate proof without a master secret key.
- A simulated proof can prove a real key as a fake key and vice versa.
- Voters can simulate a proof when coerced to reveal the real key.

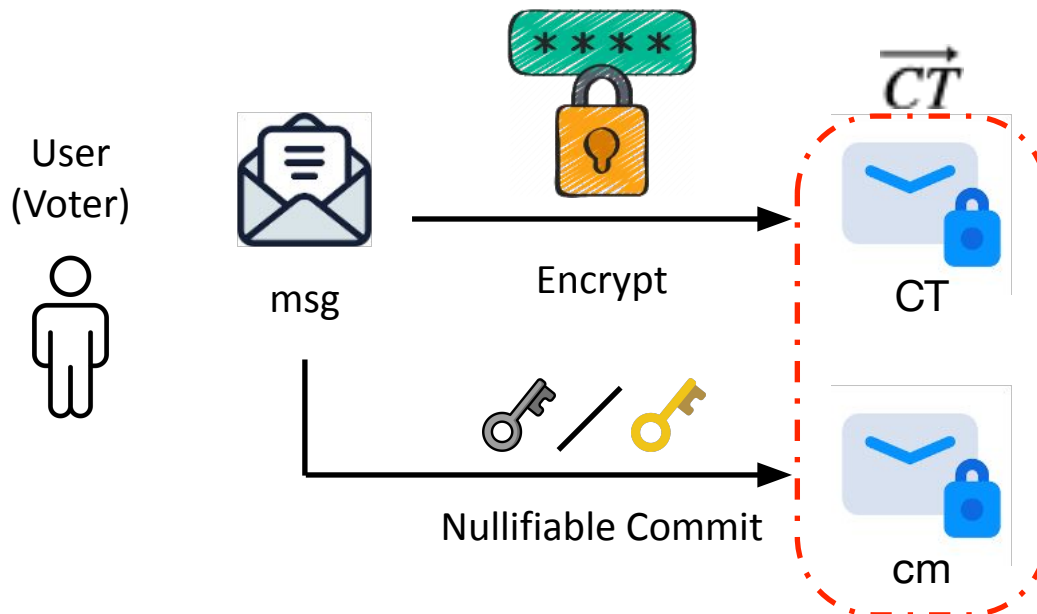
# Coercion Resistance

## ❖ Our work

- **zkVoting**: a coercion-resistant end-to-end verifiable e-voting system

## ❖ Voting phase

- Assume all voters have one real key and fake keys

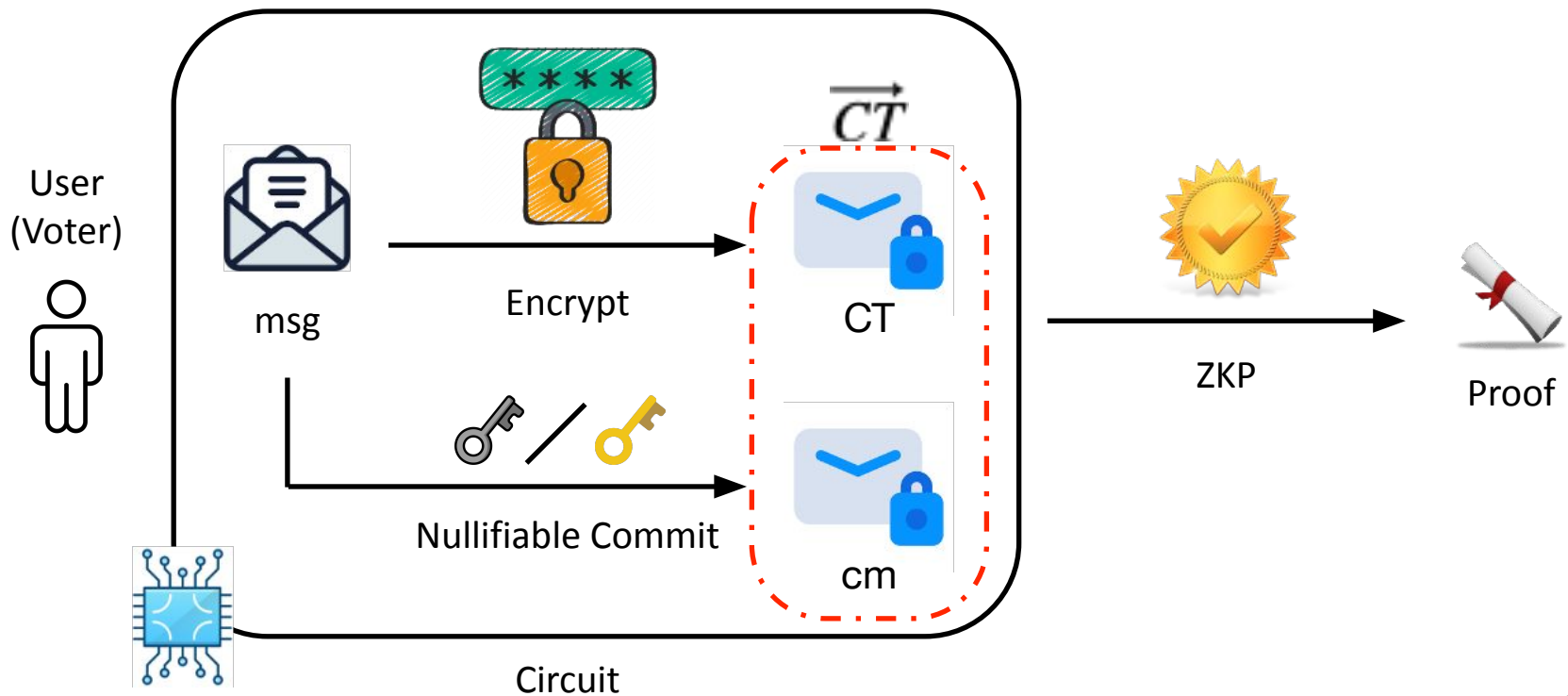


# Coercion Resistance

## ❖ Our work

- **zkVoting**: a coercion-resistant end-to-end verifiable e-voting system

## ❖ Voting phase

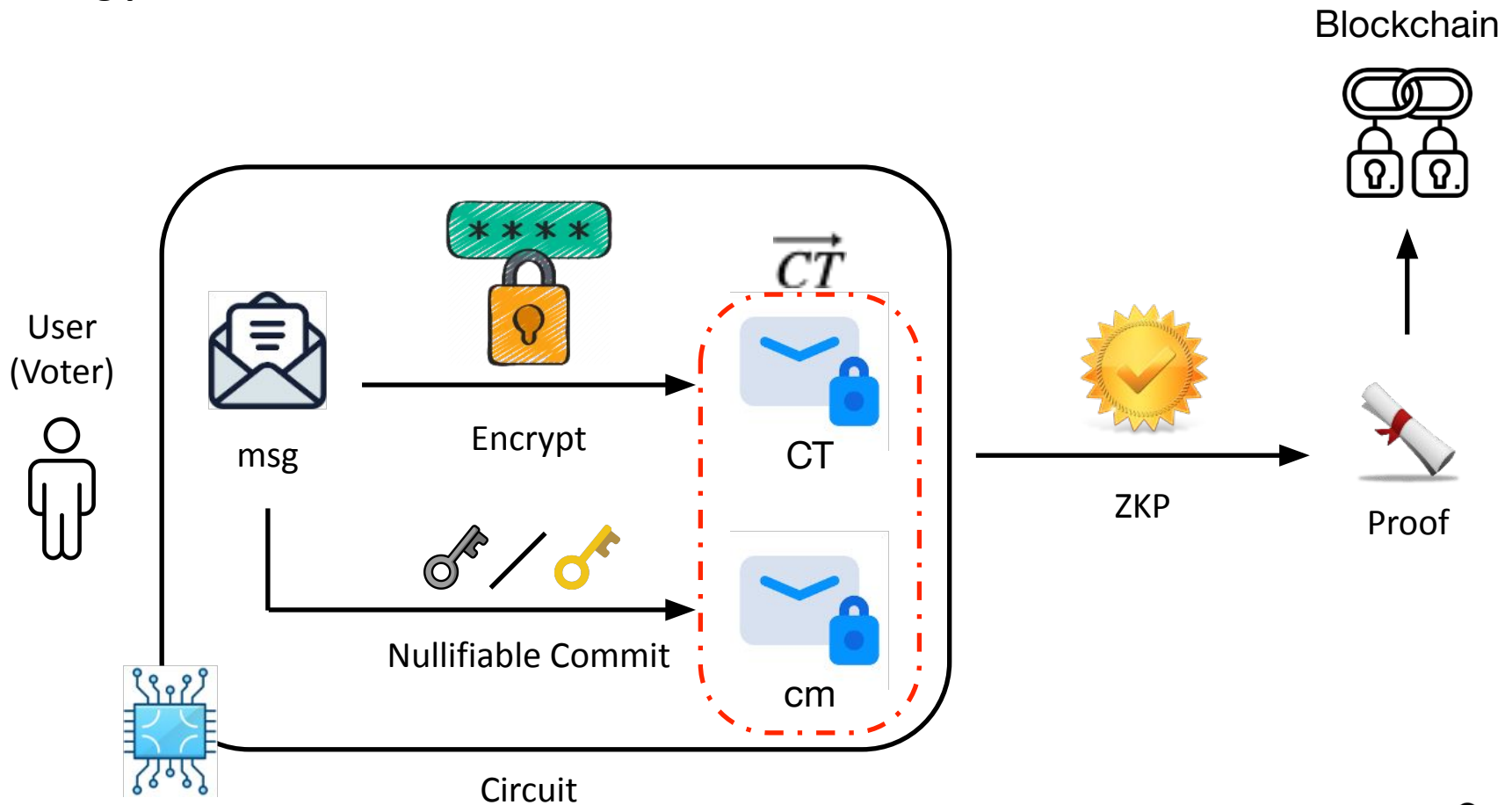


# Coercion Resistance

## ❖ Our work

- **zkVoting**: a coercion-resistant end-to-end verifiable e-voting system

## ❖ Voting phase

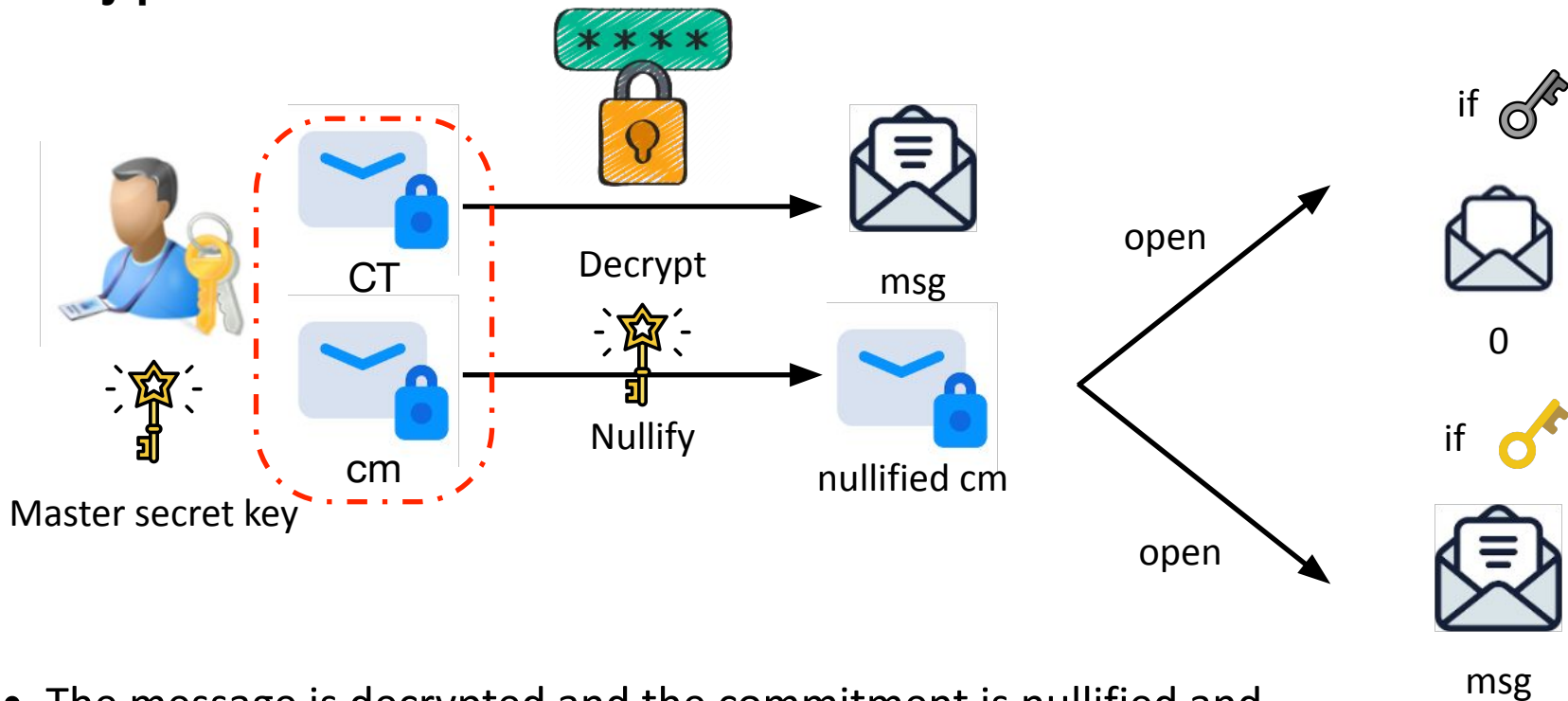


# Coercion Resistance

## ❖ Our work

- **zkVoting**: a coercion-resistant end-to-end verifiable e-voting system

## ❖ Tally phase



- The message is decrypted and the commitment is nullified and opened.

## ❖ Instantiation/Implementation

### Implementation

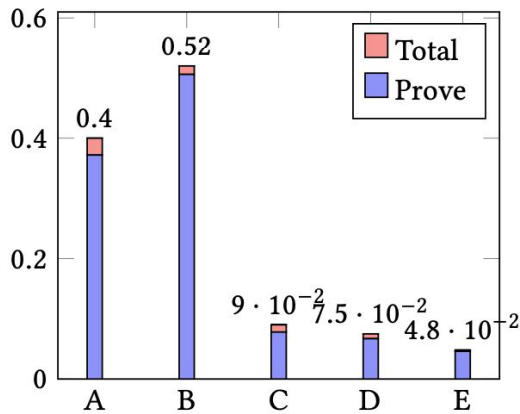
- Arkworks (Groth16)
- WASM

### Evaluations

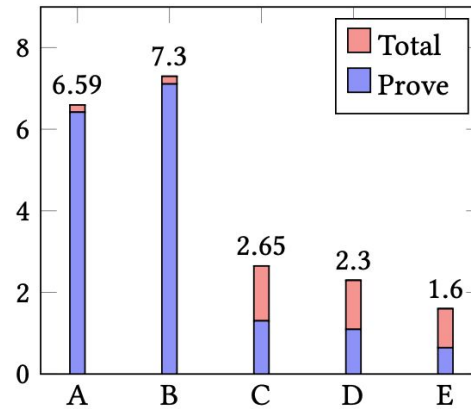
- A : Galaxy S22+
- B : Galaxy Note 10+
- C : iPhone 12 mini
- D : iPhone 14 Pro
- E : MacBook M1 Pro 16, 32GB RAM

Table 3: Voting devices

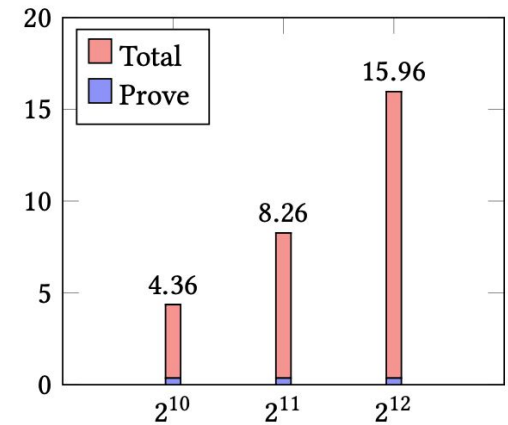
Voting devices		Specification	
A	Device1 (Galaxy S22+)	OS:	Android 13
		Processor:	Qualcomm Snapdragon 8 Gen1
		RAM:	8 GB LPDDR5 SDRAM
B	Device2 (Galaxy Note 10+)	OS:	Android 12
		Processor:	Samsung Exynos 9825
		RAM:	12 GB LPDDR4X SDRAM
C	Device3 (iPhone 12 Mini)	OS:	iOS 16.03
		Processor:	Apple A14 Bionic
		RAM:	4 GB LPDDR4X SDRAM
D	Device4 (iPhone 14 Pro)	OS:	iOS 16.03
		Processor:	Apple A16 Bionic
		RAM:	6 GB LPDDR5 SDRAM
E	Device5 (MacBook Pro 16)	OS:	macOS Ventura 13.3.1
		Processor:	Apple M1 Pro
		RAM:	32 GB



(a) Register phase



(b) Voting phase



(c) Tally phase

## ❖ Demo



---

**Thank you** for listening

Speaker: **Jihye Kim Hyunok Oh**

---



## ❖ Existing e-voting systems

### Privacy

- Ballot Privacy**: a ballot does not reveal the voter's choice
- Coercion Resistance**: a voter can cast a vote despite any influence
- Voter Anonymity**: a ballot does not reveal the identity of voters to anyone including the authority

### Verifiability

- E2E Verifiability**: a voter can identify their unique ballot and can verify all of the below
  - cast-as-intended
  - recorded-as-cast
  - tallied-as-recorded
- Eligibility Verifiability**: anyone can verify the ballot is generated only from eligible voters with a voting right and all voters cast at most one vote

## ❖ Existing coercion resistant e-voting systems

Two paradigms for coercion resistance

1. Revoting : a voter can cast a multiple ballot and only the last ballot is tallied
  - requires **another trusted party** that determines which ballot should be tallied
    - ex) tracking each voter's ballot
  - voter authentication is required during the voting phase
  - the voter requires a specific period to revote **after** being coerced
2. Fake credential : a voter can generate/use a fake credential and cast a fake ballot
  - the authority cannot check whether the voter used the correct credential
  - the voter cannot verify their vote has been recorded correctly



Fake credential + E2E verifiability

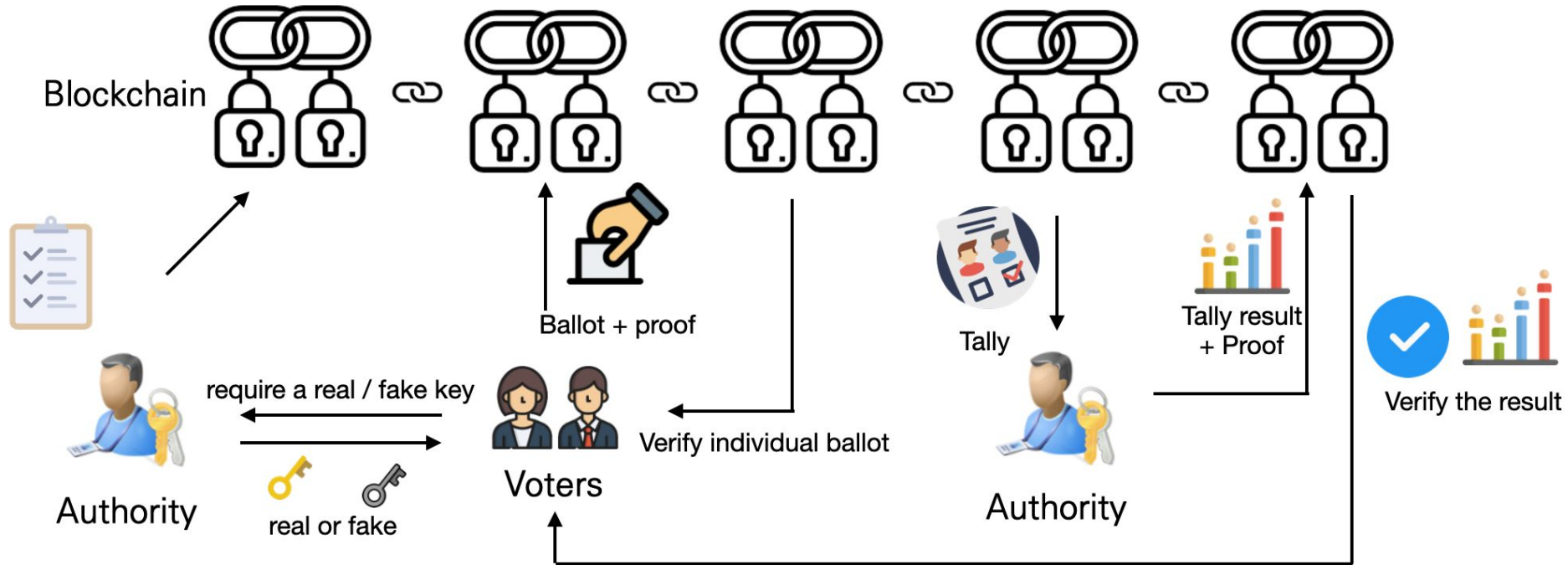
## ❖ Existing coercion resistant e-voting systems

	JCJ[1] / Civitas[2]	VoteAgain[3]	Loki[4]	zkVoting
Trusted party for Verifiability	Registrar	Registrar	Registrar	None
Trusted party for Coercion Resistance	Registrar, Tallier	Registrar, Voting server, Tallier	Voting server, Tallier	Registrar, Tallier
Paradigm	fake credentials	revoting	flexible vote updating (fake credential+revoting)	fake credentials
Time complexity for the tally phase	$n_b^2$	$n_b \log n_b$	$n_v$	$n_b$

$n_b$  : the number of ballots     $n_v$  : the number of voters

- [1] A. Juels, D. Catalano, and M. Jakobsson, "Coercion-resistant electronic elections," in Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society, 2005, pp. 61–70.
- [2] M. R. Clarkson, S. Chong, and A. C. Myers, "Civitas: Toward a secure voting system," in 2008 IEEE Symposium on Security and Privacy (sp 2008). IEEE, 2008, pp. 354–368.
- [3] W. Lueks, I. Querejeta-Azurmendi, and C. Troncoso, "VoteAgain: A scalable coercion-resistant voting system," in 29th USENIX Security Symposium (USENIX Security 20). USENIX Association, Aug. 2020, pp. 1553–1570. [Online]. Available: <https://www.usenix.org/conference/usenixsecurity20/presentation/lueks>
- [4] R. Giustolisi, M. S. Garjan, and C. Schuermann, "Thwarting last-minute voter coercion," in 2024 IEEE Symposium on Security and Privacy (SP). IEEE, 2024, pp. 115–115. [Online]. Available: <https://doi.ieeecomputersociety.org/10.1109/SP54263.2024.00112>

## ❖ Our work



## ❖ Our work

- A new cryptographic primitive, **nullifiable commitment scheme**
  - **zkVoting** : a coercion-resistant end-to-end verifiable e-voting system
- 

## ❖ Main features of zkVoting

- A voter can escape coercion by casting a vote with a fake key
- Time complexity for the tally phase is linear to the number of ballots
- Tallying one ballot takes 3.9ms in MacBook M1 Pro, 1.82x faster than Loki[4]
- Succinct time to cast a ballot (2.3s in iPhone14 Pro)

## ❖ Instantiation/Implementation

### Instantiation

Nullifiable commitment scheme: Curve25519<sup>1)</sup>

ZKP: Groth16

Hash functions: MiMC7 hash  
function

### Implementation

SNARK: libsnark, C++

Circuit: jsnark, Java

Mobile

Application

Ethereal Ganache, truffle

1) A. Kosba, Z. Zhao, A. Miller, Y. Qian, H. Chan, C. PAPAMANTHOU, R. Pass, S. ABHI, and E. SHI, “coco: A framework for building composable zero-knowledge proofs,” Cryptology ePrint Archive, Report 2015/1093, 2015. <http://eprint.iacr.org> . . . , Tech. Rep., 2015