Electronic Voting with Cryptography

Panagiotis Grontas

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών

Cryptography 25-26

Contents

Introduction

Famous words...

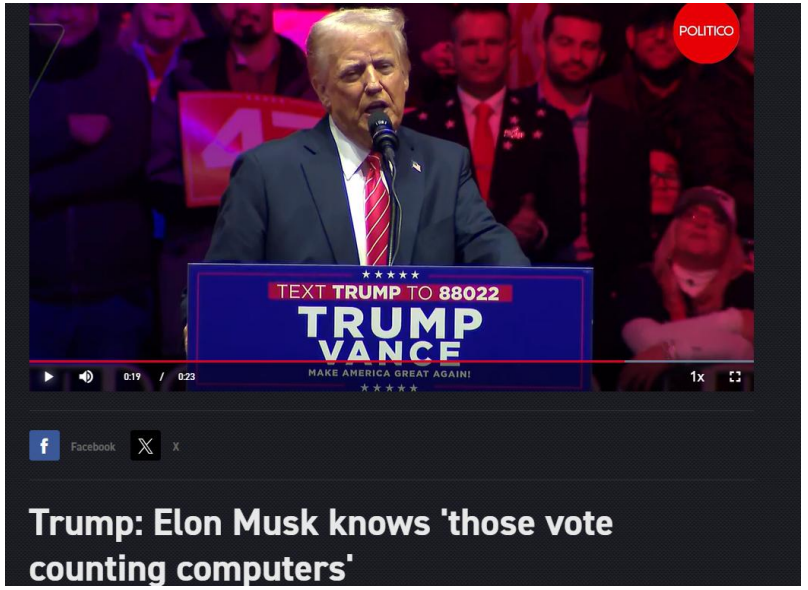
It is enough that the people know there was an election. The people who cast the votes decide nothing. The people who count the votes decide everything.

The People have spoken.... the bastards!

Famous words...

The voting booth is separated by a curtain and there is a guy behind the curtain that would write down your vote. You dictate the vote and once you ‘re done you leave, without being able to look at the ballot. Most people in their right mind, would not trust this process. The guy behind the curtain could be incompetent, hear the votes wrong and register it incorrectly or it could be that he did not like your political affiliation and prefer your vote would go to another party

Internet voting is like drunk driving…



The voting problem

Really, isn't it all about counting? What is difficult about that?

- Elections

A distributed procedure to reach a common decision

… as old as societies

… streamlined with each era's technology

… with conflicting security requirements

… where every participant is an adversary

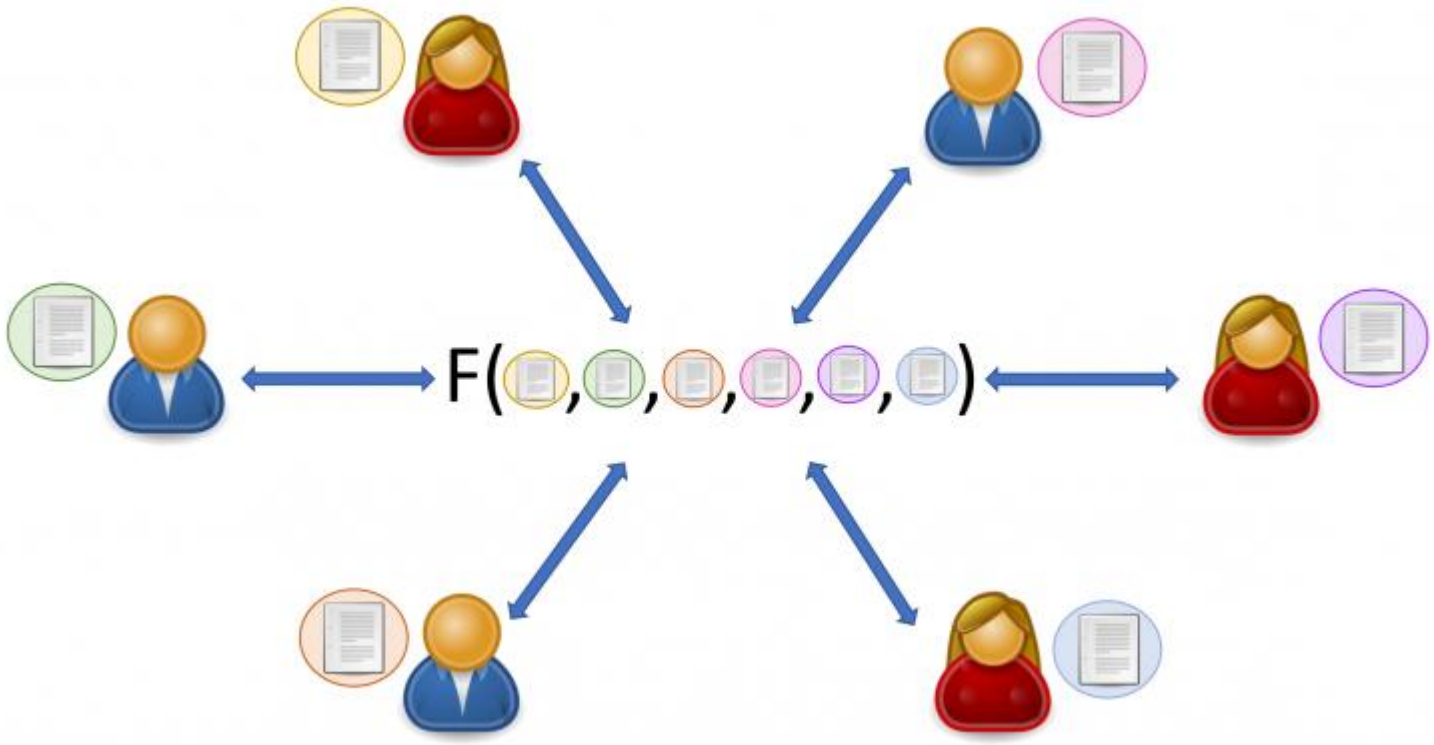
- Electronic Elections

… are already happening

  - Voter registration
  - Partial result communication and combination
  - Winner announcements

- Election only with computers
  - Inherent problems are made worse

Electronic Voting



Secure Multiparty Computation
with **stronger** security and **usability**
requirements

Security Requirements - Correctness

- Integrity
  - The result corresponds to the ballots cast
  - Not enough…
- Verifiability
  - The voter (esp. one supporting the losing side) should be convinced about integrity
  - By checking election data
  - Enables voters to regain the trust endangered by the volatile nature of computer systems and the motives of voting authorities (systemic errors or malice)

Adversary: The voting system itself

Verifiability

- Types of verifiability
  - Cast as intended
  - Recorded as cast
  - Tallied As Recorded
  - E2E Verifiability
  - Eligibility Verifiability
    - Avoid ballot stuffing

- Ways to verify
  - Individual
    - Cast as intended / Recorded As Cast
  - Universal
    - Any interested party
  - Administrative (TTP)
    - Real world elections

Verifiability ≠ Verification

Security Requirements - Privacy

- Privacy
  - The voter must express their true will
  - Secrecy
    - The vote is tied to the voter
    - The contents of the vote are never revealed
  - Anonymity
    - The vote is disassociated from the voter identity
    - Its contents can be revealed

- Adversary
  - The voting system
    - Ballot privacy
  - Voters themselves
    - Vote selling
    - Receipt Freeness
  - Other voters
    - Passive
    - Active - Coercers
    - Coercion Resistance

Privacy

- Secrecy in voting differs from secrecy in other applications (e.g., in secure messaging)
  - Ballot privacy is not absolute
  - The result leaks information
    - In a unanimous vote, everyone knows how everyone voted
    - In an all-but-one vote, the one that differs knows how everyone else voted
  - The result also yields a probability of a particular vote
    - Important in small voting populations

The primary incompatibility

- Privacy without verifiability
  - Useless
  - We don’t know if our vote will be considered
  - Leads to abstention
- Verifiability without privacy
  - Raise of hands
  - The lack of privacy forces the voters to self – censor
    - i.e., the vote loses the integrity property before it leaves the voter

Other requirements

- Fairness
  - No intermediate results are made public

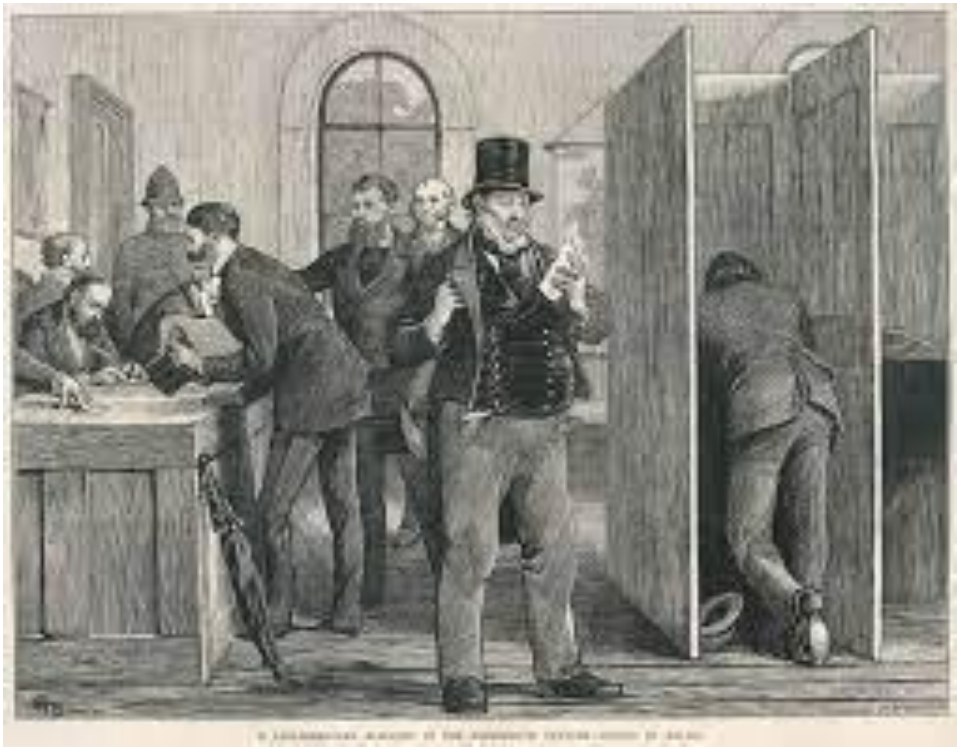
- Enfranchisement
  - The process is open to all
  - And understood by all

- Availability

- Efficiency
  - Time
  - Money

Traditional Elections: Australian Ballot

- Privacy
  - Primitive countermeasures
    - Voting in a specialized booth
    - Envelope
    - Ballot box
    - Ballot Shuffling
    - Trust in the Electoral Committee
- Verifiability
  - Only administrative!
- Integrity
  - Trust in the Electoral Committee
  - Conflicting interests
  - Trusted Third Parties



Problems in traditional elections

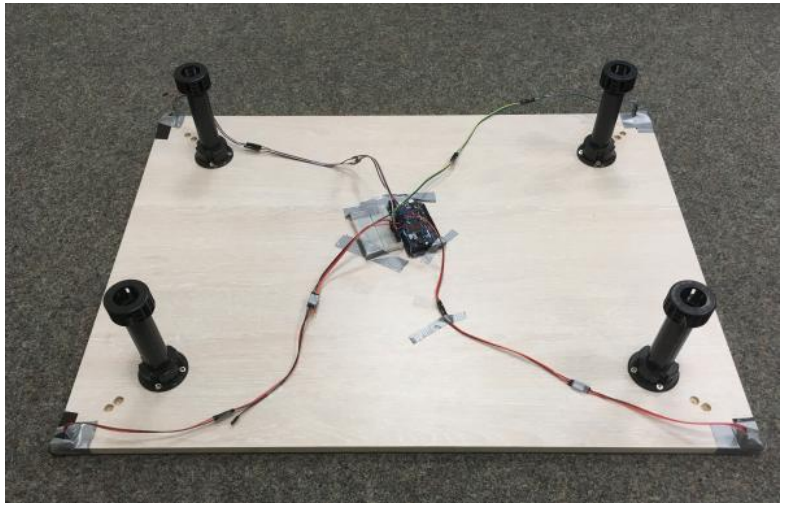
- The counting process is time-consuming.
- There are significant infrastructure expenses.
  - Ballots
  - Voting locations
  - Payment for trusted third parties
- Implementing intricate counting functions is challenging.
  - Solutions tend to raise costs
  - Elections that involve multiple rounds
- Considerations for voters with special needs.

Attacks against traditional elections: Integrity

- Before voting
  - Changing of the voter rolls
- During voting
  - Invalid ballots
  - Ballot stuffing
- During counting
  - Omit ballots
  - Cancel ballots
  - Changing Ballots
- During result announcement
  - Different result
- Countermeasures
  - Conflicting interests
  - Trusted third parties

Attacks against traditional elections: Privacy

- Incorrect ballot shuffling
  - Correlate with voting order
- Target a voter and mark their ballot
  - Different color
  - Different type of paper
- Fingerprints?
- Side channels



Countermeasures

Conflicting interests

Trusted third parties

K. Krips, J. Willemson and S. Värv, "Is Your Vote Overheard? A New Scalable Side-Channel Attack Against **Paper** Voting," *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, Stockholm, Sweden, 2019, pp. 621-634, doi: 10.1109/EuroSP.2019.00051.

Attacks against traditional elections: Vote selling – coercion resistance

- Photo of ballot
- Video of voting
  - Google glass
- Ballot switching
  - Coercer:
    - Prepare a ballot with a particular vote
  - Voter:
    - Return ballots for every candidate
- Italian (Large ballot) attack
  - Coercer:
    - You will vote for x and a particular (rare) permutation of candidates
  - Coercer:
    - Check the results for the rare permutation

Θερμός Μιχαήλ Λάμπρου	Ηλεκτρ. εσωτερικών εγκαταστ., Δομικός τεχνικών έργων, Τεχν. επεξ. ύδατος & αποβλήτων, Πρόεδ. Αγροτοβ/κού Συνετ. Εγκλημενού	Δ. Ε. ΕΛΛΟΜΕΝΟΥ
Καββαδά Σαπφώ Σωκράτη	Συνταξιούχος εκπαιδευτικός	Δ. Ε. ΕΛΛΟΜΕΝΟΥ
Καγκελάρης Γεράσιμος (Τούρκος) Αθανασίου	Αγρότης	Δ. Ε. ΑΠΟΛΛΩΝΙΩΝ
Καράμπαλης Φίλιππος Αναστασίου	Ιδιωτικός υπάλληλος	Δ. Ε. ΛΕΥΚΑΔΑΣ
Κατηφόρης Χρήστος Φωτίου	Πρώην Αντιδήμαρχος Απολλωνίων, Πολιτικός μηχανικός	Δ. Ε. ΑΠΟΛΛΩΝΙΩΝ
Κατωπόδη Ρωξάνη Ηλία	Οικονομολόγος, Ιδιωτ. Υπάλληλος	Δ. Ε. ΚΑΡΥΑΣ
Κατωπόδης Σοφοκλής Χρήστου	Συνταξιούχος	Δ. Ε. ΚΑΡΥΑΣ
Κηρολίβανος Πανταζής Ιωάννη	Πολιτικός μηχανικός, Πρόεδρος Ν.Ε. ΤΕΕ Λευκάδας	Δ. Ε. ΛΕΥΚΑΔΑΣ
Κονδυλάτου Νικολέττα Ιωάννη	Καθηγήτρια Φυσικής Αγωγής	Δ. Ε. ΛΕΥΚΑΔΑΣ
Κορφιάτης Παναγιώτης Κωνσταντίνου	Πρόεδρος Κοινότητας Αθανίου, Συνταξιούχος	Δ. Ε. ΑΠΟΛΛΩΝΙΩΝ
Κούρτη Βασιλική Αθανασίου	Ιδιωτική υπάλληλος	Δ. Ε. ΛΕΥΚΑΔΑΣ
Κούρτη Ελισάβετ (Ελσα) Ευθυμίου	Οικονομολόγος, Δημόσιος υπάλληλος	Δ. Ε. ΛΕΥΚΑΔΑΣ
Κοψιδά Θεοδώρα (Δώρα) Θωμά	Ιδιωτική υπάλληλος	Δ. Ε. ΛΕΥΚΑΔΑΣ
Κτενά Ιωάννα Θεοδώρου	Γεωλόγος	Δ. Ε. ΛΕΥΚΑΔΑΣ
Κωστόπουλος Γεώργιος Ευαγγέλου	Αξιωματικός Ενόπλων Δυνάμεων ε.α.	Δ. Ε. ΛΕΥΚΑΔΑΣ
Μάλλιου Βαρβάρα Αθανασίου	Δημόσιος υπάλληλος	Δ. Ε. ΛΕΥΚΑΔΑΣ
Μανωλίτσης Βελισσάριος Ζώη	Μηχανικός ηλεκτρονικών υπολογιστών & Πληροφορικής	Δ. Ε. ΛΕΥΚΑΔΑΣ
Μελάς Ιωάννης Χαραλάμπους	Ελεύθερος επαγγελματίας	Δ. Ε. ΛΕΥΚΑΔΑΣ
Μεσσήνης Ιωάννης Γεωργίου	Χωματουργικές εργασίες	Δ. Ε. ΕΛΛΟΜΕΝΟΥ
Μήτσουρα Σταυρούλα Διονυσίου	Ιδιωτική υπάλληλος	Δ. Ε. ΕΛΛΟΜΕΝΟΥ
Μήτσουρας Εμμανουήλ-Αθανάσιος Ηλία	Καθηγητής μουσικής, Αρχιμουσικός	Δ. Ε. ΛΕΥΚΑΔΑΣ
Μιχαλάτος Κων/νος Ευσταθίου	Ελεύθερος επαγγελματίας	Δ. Ε. ΛΕΥΚΑΔΑΣ
Νικητάκης Μάρκος Βασιλείου	Πρώην Αντιδήμαρχος, Μαθηματικός, Πρώην εργαζόμενος Υπουργείου Οικονομικών	Δ. Ε. ΛΕΥΚΑΔΑΣ

First generation electronic voting

- In reality
  - Replace the ballot box with a computer
  - Input the voter choice
  - Electronic counting
  - No secrecy whatsoever!
- Voting with an untrusted intermediary
  - Malicious software
  - Programming errors
  - Targeted attacks
  - Interface problems
- No verifiability
- Open source: Necessary but not sufficient

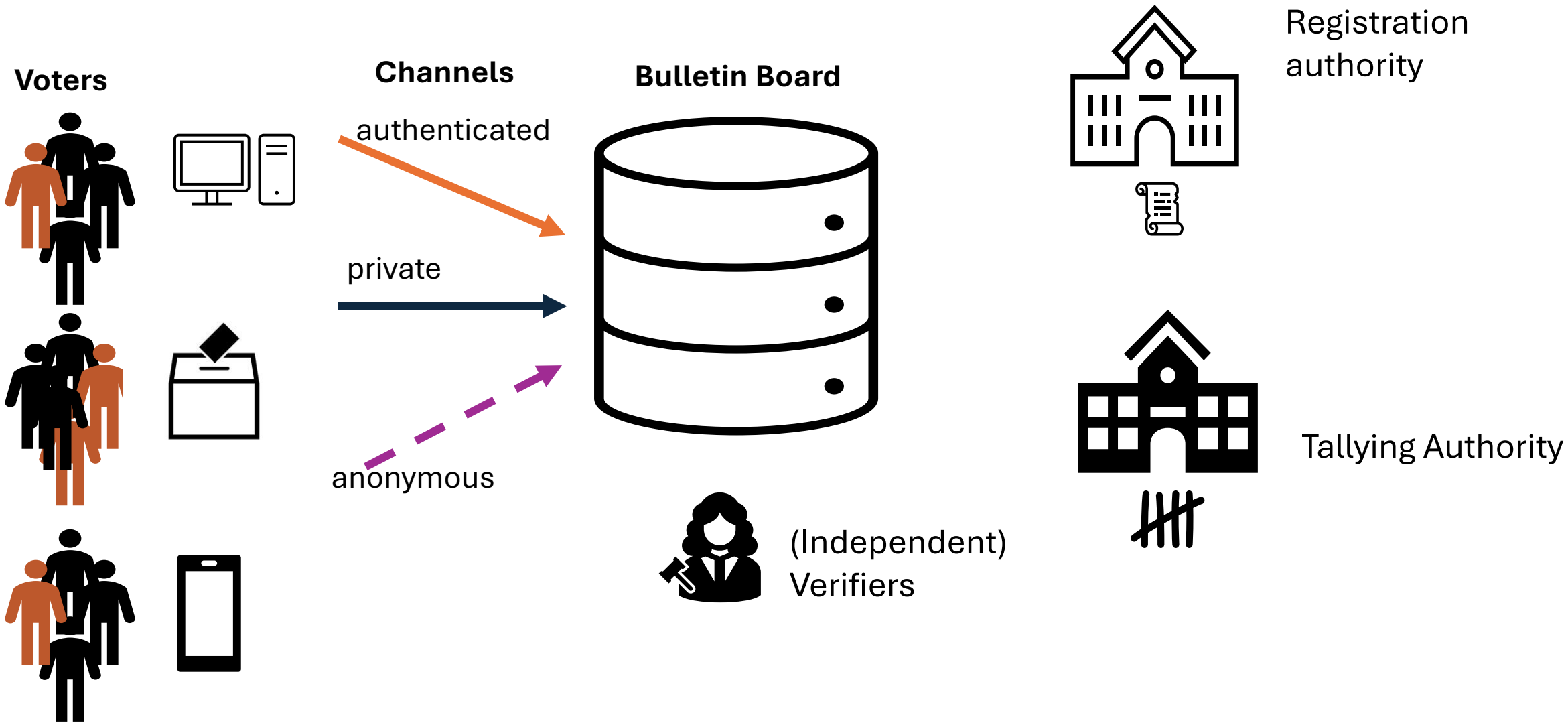
First generation electronic voting

- Software independence (Rivest)
  - *A voting system is software-independent if an undetected change or error in its software cannot cause an undetectable change or error in an election outcome.*
- **Solution 1^n:**
  - Voter Verifiable Paper Trail (VVPAT) + Risk Limiting Audits (RLA)
- **Solution 2^n:**
  - Cryptography!

Second generation electronic voting

- Elections without TTPs
- Cryptography
  - Secrecy
  - Integrity
  - Verifiability
- Basic Ideas
  - David Chaum (1981)
  - Josh Benaloh (1987)
  - Ben Adida (2008)
  - Cramer, Gennaro, Schoenmakers (1997)
  - Juels, Catalano, Jakobsson (2005)

General Architecture



Bernhard, M. *et al.* (2017). Public Evidence from Secret Ballots. In: Krimmer, R., Volkamer, M., Braun Binder, N., Kersting, N., Pereira, O., Schürmann, C. (eds) Electronic Voting. E-Vote-ID 2017. Lecture Notes in Computer Science(), vol 10615. Springer, Cham. https://doi.org/10.1007/978-3-319-68687-5_6

Cryptographic Voting Schemes

Architecture and Primitives

Public Key Cryptosystems

- ElGamal Encryption
  - $\mathbb{G}$ is a cyclic group of prime order q generated by g
  - $sk \xleftarrow{\$} \mathbb{Z}_q, pk = g^{sk}$
  - pk belongs to the tallying authority
  - $Enc_{pk}(m) = (g^r, m \cdot pk^r), r \xleftarrow{\$} \mathbb{Z}_q, m \in \mathbb{G}$
  - $Dec_{sk}(c) = c_2 \cdot c_1^{-\mathrm{sk}} = \mathrm{m}$
- Exponential ElGamal
  - $Enc_{pk}(m) = (g^r, g^m \cdot pk^r), r \xleftarrow{\$} \mathbb{Z}_q, m \in \mathbb{Z}_q$
  - $Dec_{sk}(c) = c_2 \cdot c_1^{-\mathrm{sk}} = \mathrm{g^m}$
  - Solve 'small' DLOG

- Homomorphic properties

$$Enc_{pk}(v_1) \otimes Enc_{pk}(v_2) =$$

$$(g^{r_1}, g^{v_1} \cdot pk^{r_1}) \otimes (g^{r_2}, g^{v_2} \cdot pk^{r_2}) =$$

$$(g^{r_1+r_2}, g^{v_1+v_2} \cdot pk^{r_1+r_2}) = Enc_{pk}(v_1 + v_2)$$

- Reencryption

$$ReEnc_{pk}(c) = c \otimes Enc_{pk}(1) =$$

$$(g^r, m \cdot pk^r) \otimes (g^{r_1}, pk^{r_1}) = (g^{r\ +r_1}, m \cdot pk^{r\ +r_2})$$

Alternatives: Paillier Cryptosystem, DJ Cryptosystem

- Paillier, Pascal (1999). "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes" (PDF). EUROCRYPT '99.
- Ivan Damgård, Mads Jurik: A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System. Public Key Cryptography 2001: 119-136

Benaloh Challenge

A *cut & choose* technique to encrypt a vote from an untrusted device

- The voter enters the choice to the device
- The device creates the ciphertext
- The voters selects Audit or Cast
- On Audit
  - The device releases the randomness used to encrypt the choice
  - The voter can recreate the encryption on their own
  - The encrypted vote is not admissible
  - Repeat
- On Cast
  - The ballot is sent to the **BB**

Wojciech Jamroga:
Pretty Good Strategies for Benaloh Challenge. E-Vote-ID 2023: 106-122

Basic Idea:

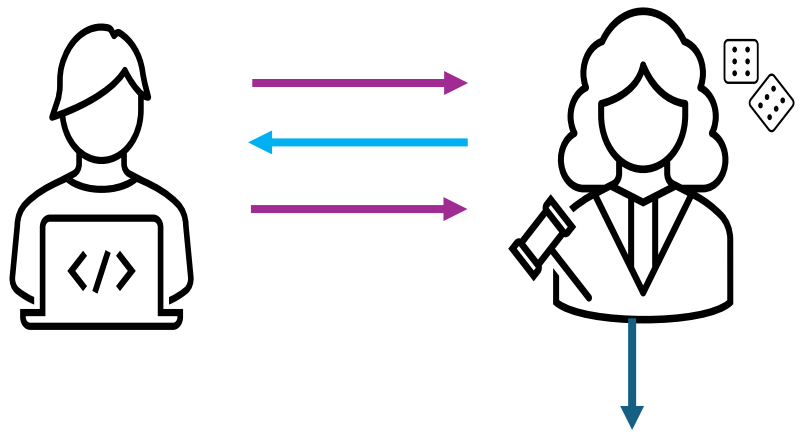
- The device does not know in advance if the voter will audit or cast
- If it changes the voter input it might be caught
- Game theoretic argument

Commitment schemes

- **Pedersen Commitments**
- $\mathbb{G}$ is a cyclic group of prime order q generated by g, h
  - $Commit(m) = g^m h^r$
  - $Open(c, m, r) = (g^m h^r =_? c)$
- Perfectly hiding
- Binding if DLOG is hard
- If $DLOG_g(h) = x$ is known:
  - $m,\ m + x(r - r')\ mod\ q$ have the same commitments under r, r'
- Trusted setup!

- **Generalized Form**
- Commitment to a vector $\mathbf{m} = (m_1, \ldots, m_n)$
- $\mathbb{G}$ is a cyclic group of prime order q generated by $g_1, \ldots, g_n, h$
  - $Commit(\mathbf{m}) = h^r \prod_i g_i^{m_i}$

Schnorr's Protocol

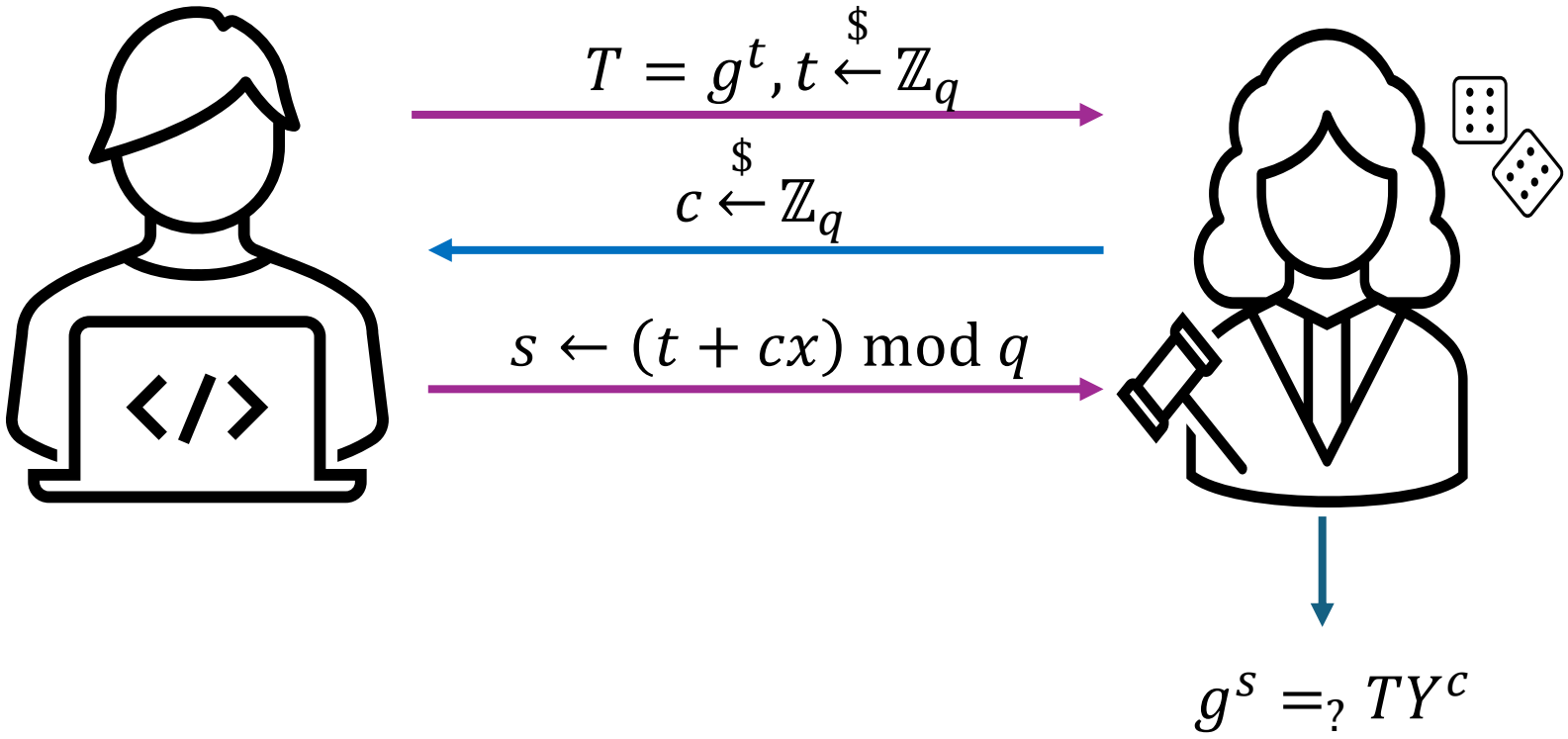


- Proof of Knowledge of a Discrete Logarithm
- $PoK\{x: g^x = Y : Y, g \in \mathbb{G}\}$
- Public Input
  - $\mathbb{G}$ is a cyclic group of prime order q generated by g
  - A group element $Y \in \mathbb{G}$
- Witness
  - $x \in \mathbb{Z}_q$

Schnorr, C. P. (1991). "Efficient signature generation by smart cards". Journal of Cryptology. 4 (3): 161–174. doi:10.1007/BF00196725. S2CID 10976365.

Schnorr's Protocol (II)

$$PoK\{x : g^x = Y : Y, g \in \mathbb{G}\}$$



Non-interactive Schnorr (**DLPRV**)

- Public input: $g \in \mathbb{G},\ ord(\mathbb{G}) = q, Y \in \mathbb{G}$
- Private input: $x \in\ \mathbb{Z}_q : Y = g^x$

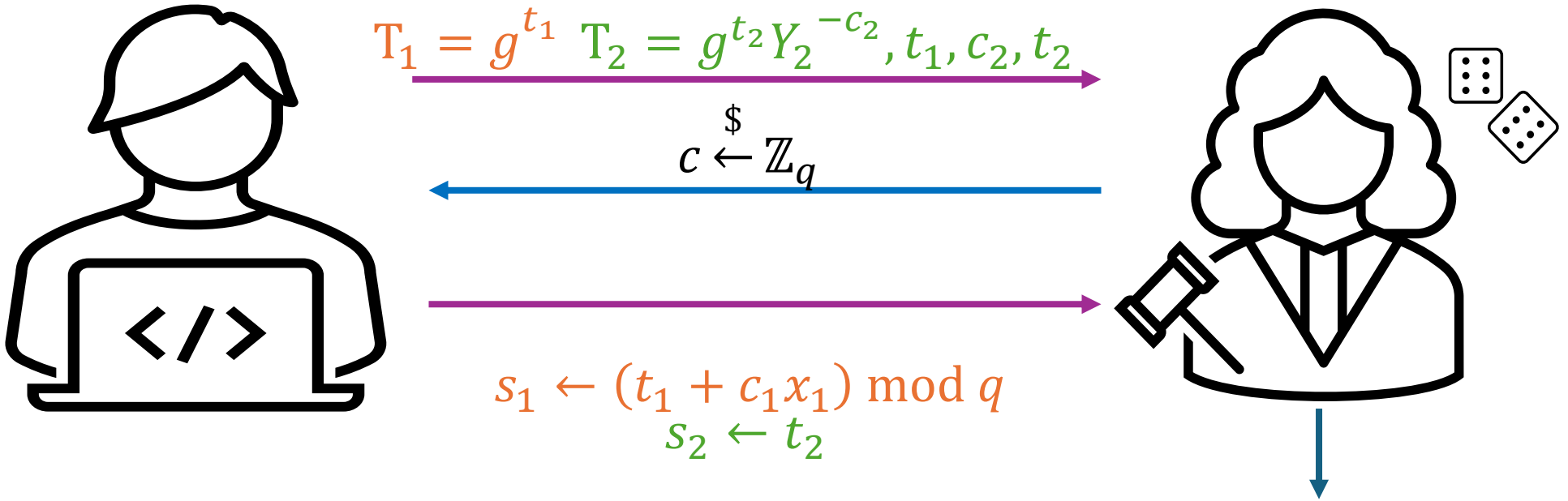
$\boldsymbol{DLPRV}(x, g, Y)$

- Select $t \xleftarrow{\$} \mathbb{Z}_q$ and compute $T = g^t$
- Compute $\boldsymbol{c \leftarrow H(g, Y, T)}$
- Compute $s \leftarrow t + cx$
- The proof is: $\pi = (c, s)$
- **DLVF:** Public verifiability by checking if $c = H(g, Y, g^s Y^{-c})$

$\boldsymbol{DLVF}(g, Y, \pi)$

OR Schnorr (**DJPRV**)

- Proof of knowledge of one out of two DLOGs
- $PoK\{(\boldsymbol{x_1}, x_2): g^{\boldsymbol{x_1}} = Y_1 \textbf{ OR } g^{x_2} = Y_2,\ Y_1, Y_2, g \in \mathbb{G}\}$



DJVF $g^{s_1} =_? T_1 Y_1^{c_1}$ and $g^{s_2} =_? T_2 Y_2^{c_2}$
and $c = c_1 + c_2$

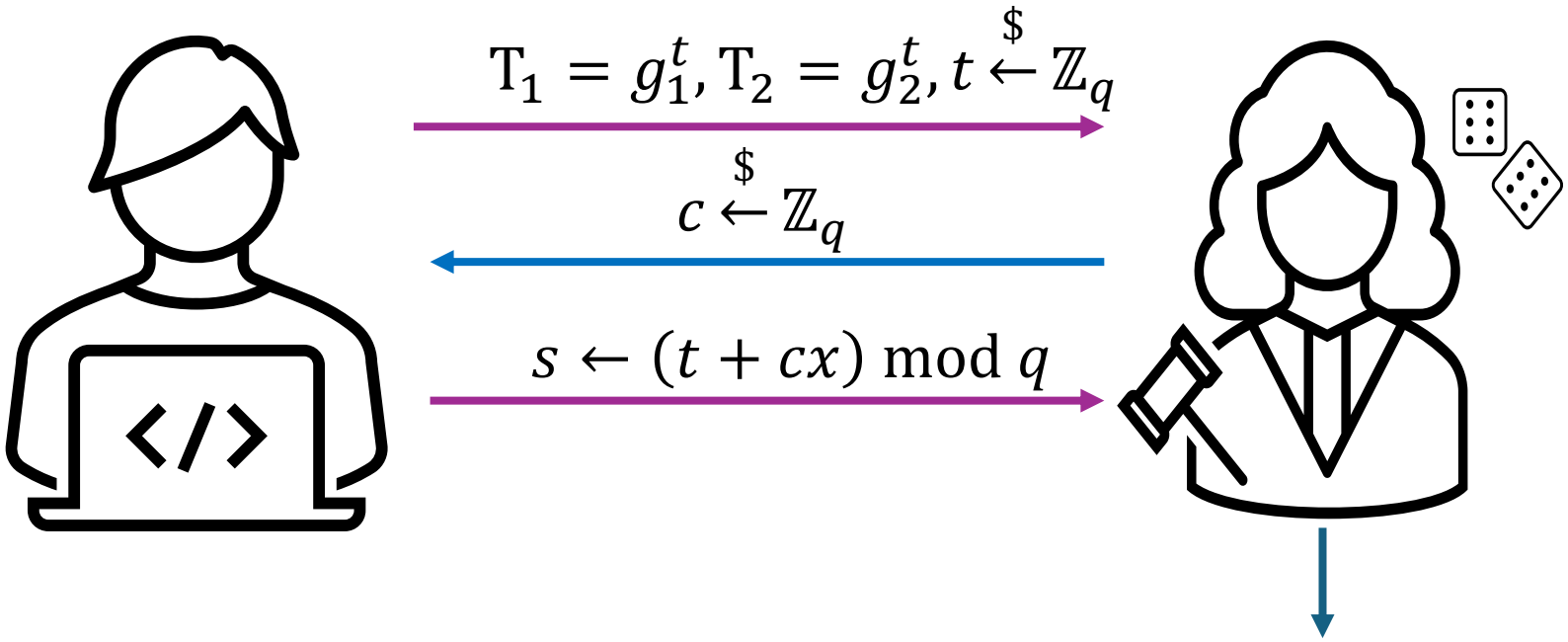
Pitfalls of the Fiat-Shamir Heuristic

- **Weak FS:** Input to hash function contains only commitment
  - $c \leftarrow H(T)$
- **Strong FS:** Input to hash function contains commitment, statement to be proved and all public values generated so far.
  - $\boldsymbol{c \leftarrow H(g, Y, T)}$
- If the prover is allowed to select their statement adaptively then the weak FS yields unsound proofs
- Proofs created using the weak FS have implications to the privacy and verifiability of Helios and other similar voting systems.

Bernhard, Pereira, Warinschi (2012) How Not to Prove Yourself: Pitfalls of the Fiat-Shamir Heuristic and Applications to Helios. *ASIACRYPT 2012*

Chaum – Pedersen protocol (**EQPRV**)

- Proof of knowledge and equality of two DLOGs
- $PoK\{x: g_1^x = Y_1, g_2^x = Y_2 : Y_1, Y_2, g_1, g_2 \in \mathbb{G}\}$



Enc+PoK for non-malleability

- Malleability:
  - The ability to transform a valid ciphertext into another (meaningfully related) valid ciphertext without decrypting and encrypting again
- To achieve non malleability the $Enc + PoK$ construction may be used:
  - Append a NIZK PoK of *the encryption randomness* to the ciphertext
- In ElGamal for instance
  - $Enc_Y(m) = (g^r, m \cdot Y^r, c, s)$ where $(c, s) = \boldsymbol{DLPRV}(r, g, g^r)$
  - Before decrypting check if $\boldsymbol{DLVF}\big(g, g^r, (c, s)\big) = 1$
  - Recall that $c = H(g, g^r, g^s(g^r)^{-c})$

Plaintext equivalence test (**PET**)

Do two ciphertexts c, c' encrypt the same plaintext?

$$\boldsymbol{c} = (c_1, c_2) = Enc_{pk}(m), \quad \boldsymbol{c'} = (c'_1, c'_2) = Enc_{pk}(m')$$

$$\boldsymbol{c_{PET}} = \frac{\boldsymbol{c}}{\boldsymbol{c'}}$$

$$\boldsymbol{c_{PET,i}} = \left(\frac{\boldsymbol{c}}{\boldsymbol{c'}}\right)^{z_i} \quad \pi_{i1} = EQPRV(z_i)$$

$$\phi = \prod \boldsymbol{c}_{PET,i} = (x, y)$$

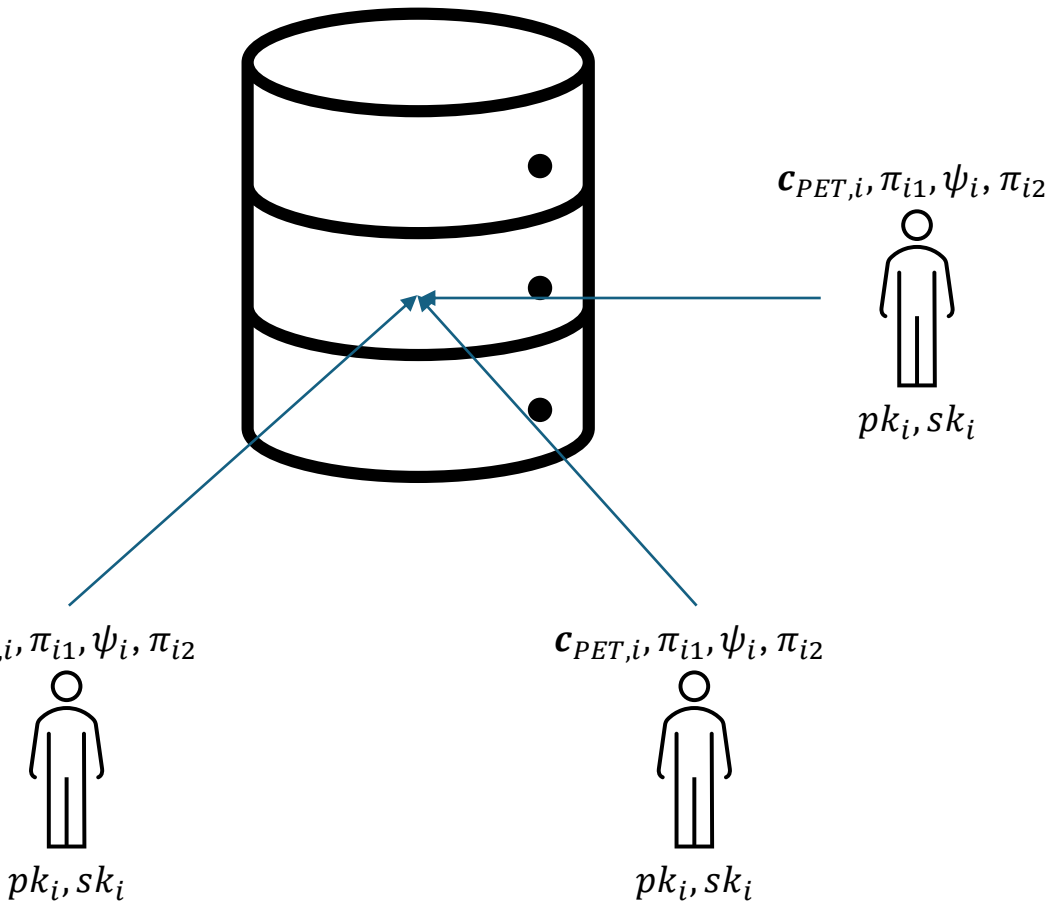
$$\psi_i = x^{sk_i} \quad \pi_{i2} = DLRPV(sk_i)$$

$$\rho = y / \prod \psi_i$$

$$\rho =_? 1$$

pk_i, sk_i

$pk = \prod pk_i$



Jakobsson, M., Juels, A. (2000). Mix and Match: Secure Function Evaluation via Ciphertexts. In: Okamoto, T. (eds) Advances in Cryptology — ASIACRYPT 2000

Helios

Helios' Facts

- Elections in the browser
  - Open-Audit: Everyone has access to all election data for verifiability
  - **Trust no one for integrity – trust the server for privacy**
  - Low coercion environments
- 2.000.000 votes cast so far
  - ACM, IACR and university elections
  - Can be used online https://vote.heliosvoting.org/ or deployed locally
- Based on:
  - Verifiable mixnets – Helios 1.0 (Sako-Killian, Eurocrypt 95)
  - **Homomorphic tallying – Helios 2.0 (Cramer-Genaro-Shoenmakers, Eurocrypt 97)**
  - **Benaloh Challenge**
- Many variations
  - Belenios (Helios-C)
  - Zeus

Ben Adida. 2008. Helios: web-based open-audit voting. In Proceedings of the 17th conference on Security symposium (SS'08). USENIX Association, USA, 335–348.

Participants

- **Election administrator:** Create the election, add the questions, combine partial tallies
- **BB - Bulletin' Board**: Maintain votes (**Ballot Tracking Center**) and audit data
- **TA - Trustees (Talliers)**: Partially decrypt individual (in Helios 1.0) or aggregated (in Helios 2.0) ballots
- **RA - Registrars (Helios-C):** Generate cryptographic credentials for voters
- $EA = (RA, TA, BB)$
- **Eligible voters** optionally identified by random alias or external authentication service (Google, Facebook, LDAP)
  - Authenticated channel between voter and BB (username, password)

Auditing Process

- Individual Verifiability
  - Cast as intended
    - After ballot creation (encryption) but before authentication, each voter can choose if they will audit or cast the ballot.
    - **On audit:** Helios releases the encryption randomness and the voter can recreate the ballot using software of their choice.
    - An audited ballot cannot be submitted.
  - Recorded as cast
    - Each encrypted ballot and related data are hashed to a tracking number.
    - Every voter can check if the assigned number exists in the Ballot Tracking Center (BTC).

Auditing Process

- Universal Verifiability
  - Tallied as recorded - Every interested party may
    - Retrieve ballots from BTC
    - Compare identities with eligible voters (if applicable)
    - Recompute tracking numbers
    - Aggregate the ballots and check equality with official encrypted tally before decryption
  - Verify decryption proofs

Formal Description: Setup

- Executed by the Election Administrator
- Creates cryptographic groups, defines message space etc.
- Reusable for many elections

$$Setup(1^{\lambda}) = \begin{cases} \mathbb{G}, q, g \\ H_q : \{0,1\} \rightarrow \mathbb{Z}_q \\ (\boldsymbol{DLPRV}(x, g, Y), \boldsymbol{DLVF}(g, Y, \pi)) \\ (\boldsymbol{EQPRV}(x, g_1, Y_1, g_2, Y_2), \boldsymbol{EQVF}(g_1, Y_1, g_2, Y_2, \pi)) \\ (\boldsymbol{DJPRV}(x_1, x_2, g, Y_1, Y_2), \boldsymbol{DJVF}(g, Y_1, Y_2, \pi)) \\ BB \leftarrow \emptyset \end{cases}$$

Formal Description: SetupElection

- The members of the TA cooperate to create their **joint** public key
  - Compute member key pair: $sk_i \xleftarrow{\$} \mathbb{Z}_q, pk_i \leftarrow g^{\mathrm{sk}_i}$
  - Publish $pk_i, DLPRV(sk_i, g, pk_i)$
  - Compute election public key: $pk \leftarrow \prod_i pk_i$
- Create list of eligible voters V_l
- Create list of candidates $CS = \{0,1\}$ (for simplicity)
- Publish everything into BB
  - $BB \Leftarrow \{pk_i, pk, V_l, CS\}$

Formal Description: Voting

Vote(i,v):

$v \in \{g^0, g^1\}$

$Enc_{\mathrm{pk}}(g^v) \rightarrow (g^r, g^v \cdot \mathrm{pk}^r) = (R, S)$

$EQPRV(r, g, R, pk,\ S)$ ***OR*** $EQPRV(r, g, R, pk, Sg^{-1}) \rightarrow \pi_V$

$\mathrm{b} = (R, S, \pi_V)$

Valid(i,b):

Return 1 if $i \in V_l$ and $EQVF(\pi_V) = 1$

Append(I,b):

$BB \leftarrow (i, b)$ if $Valid(b) = 1$

VerifyVote(i,b,BB):

Return 1 if $b \in BB$ **and** $Valid(i, b) = 1$

Publish(BB):

Return $\mathrm{PBB} = \{b\}$ i.e. remove id's from ballots and keep one ballot per voter id

Occurs after all voters have voted

Formal Description: Tally

Tally(PBB, sk_i):

Validate all proofs in PBB

Compute $(R_\Sigma, S_\Sigma) \leftarrow \prod b$ for all $b \in PBB$

Distributed Decryption of $(R_\Sigma, S_\Sigma) \rightarrow g^t$

Each TA_i

posts $\left(D_i = R_\Sigma^{sk_i}, EQPRV(sk_i, g, pk_i, R_\Sigma, D_i)\right)$

computes $\frac{S_\Sigma}{\prod_i D_i} \rightarrow g^t$

solves small DLOG to get t

posts $\pi_T = EQPRV(sk_i, g, pk_i, R_\Sigma, S_\Sigma \cdot g^{-t})$

Formal Description: Verify

Verify(BB,PBB, t, π_T):

Check correct construction of PBB

- Only last ballot kept
- All kept ballots belong to eligible voters
- All kept ballots had valid proofs

Recompute $(R_\Sigma, S_\Sigma) \leftarrow \prod b$ for all $b \in PBB$

Verify π_T

JCJ and CIVITAS

Coercion Resistance

- A stronger adversary
- Active attacks
  - Vote for a specific candidate
  - Vote randomly
  - Completely abstain from voting
  - Yield private keys – allow simulation
  - Monitor voting systems
- The essential security property for Internet voting
- **Note:** Coercion Resistance ⇒ Receipt freeness

The JCJ coercion resistance framework

- Intuition:
  - The adversary will not coerce, if they cannot verify that the coercion attempt will succeed
- Techniques
  - Each voter can vote multiple times
  - The voter can generate and register credentials (=random group elements)
  - There is a single valid credential for each voter (=the one registered)
  - During voting the voter may generate indistinguishable credentials through a device or some other manner
  - All the votes accompanied with other credentials are considered fake and should not be counted

A. Juels, D. Catalano, and M. Jakobsson, "Coercion-resistant electronic elections," in Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society, WPES, 2005

JCJ Assumptions

- Each voter has a moment of privacy where they can cast their real ballot
  - May occur before / after the adversarial attack
- The casting phase is anonymous
  - Otherwise, the forced abstention attack would always succeed
- The coercer is uncertain about the *behavior of all the voters*
  - If everyone else votes, then the abstention attack will always succeed
  - If nobody votes for the candidate the coercer demands then the attack will succeed
  - Insertion of dummy votes
- Untappable registration
  - Or the coercer becomes the voter

JCJ Workflow

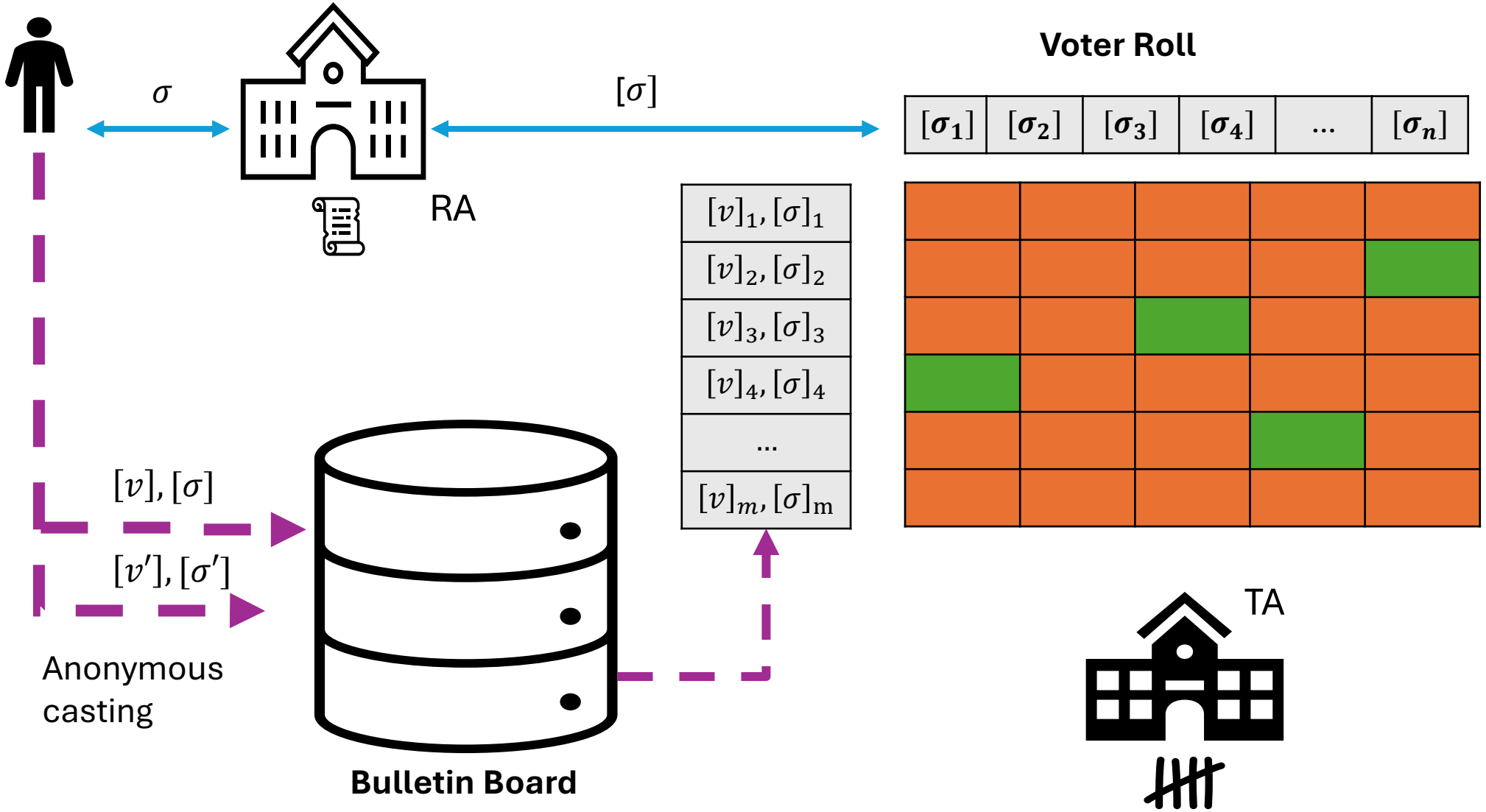
- The voter registers their real credential
  - Untappable registration – occurs once but may be reused
  - The voter may create the credential either alone or together with an authority
- The authorities publishes all real credentials in encrypted form
  - Voter roll
- Coercion Attack
  - The voter generates a fake but indistinguishable credential
  - The voter complies with the commands of the coercer
  - The coercer may monitor the voter afterwards, except during…
- Moment of privacy
  - The voter casts their vote of choice accompanied with their real credential

JCJ Workflow (2)

- Tallying
  - The BB is anonymized
  - Ballot weeding
  - The authorities disregard **in a verifiable manner**:
    - all duplicate ballots (e.g. by keeping only the last ballot per voter)
    - all ballots with fake credentials
  - How: Blind credential comparisons using PET
    - Between all ballots
    - Between unique ballots and the voter roll
    - Proof of PET serves verifiability

The scheme

M. R. Clarkson, S. Chong, and A. C. Myers, “Civitas: Toward a Secure Voting System,” in S&P’08. IEEE, 2008.



JCJ Discussion

- **Quadratic tallying time**
  - $O(m^2)$: To keep one ballot per credential
  - $O(mn)$: To filter out ballots with fake credentials
  - Goal: $O(m + n)$
  - **Solutions:**
    - Blinded hashing
    - Anonymity sets
    - Structured credentials

- **Difficult to use by the voters**
  - Need for hardware tokens that generate fake credentials
  - Solutions:
    - Panic passwords
    - Except for valid, invalid a password can signal coercion
    - Panic password list per voter
    - Moment of privacy:
      - Valid password
    - Coercion:
      - Panic password