Electronic Voting with Cryptography

Panagiotis Grontas

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών

Advanced Topics In Cryptography
Προχωρημένα Θέματα Κρυπτογραφίας
Μάρτιος 2025

Contents

- Introduction
  - Problem Definition
  - Requirements
  - System Model
- Cryptographic primitives
- Voting paradigms
  - Homomorphic Encryption
  - Mixnets
  - Blind/Ring Signatures
  - Decentralized Voting
- Systems and Attacks
- Security Analysis
  - Formal Definitions
    - Verifiability
    - Privacy
    - Receipt-Freeness
    - Coercion resistance
    - Everlasting privacy
  - Security Proofs

Introduction

Famous words…

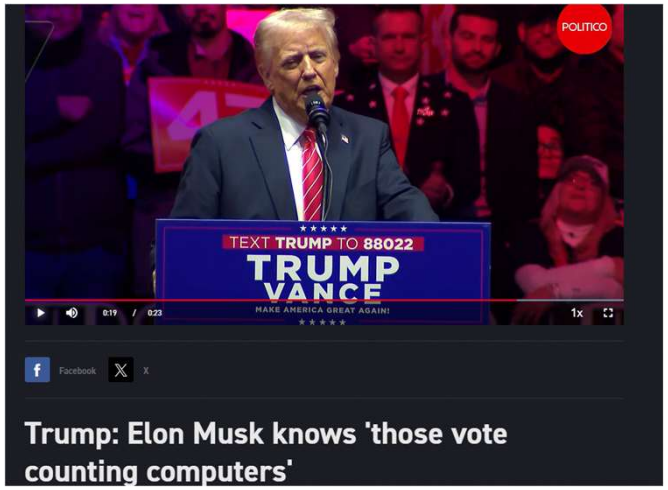
It is enough that the people know there was an election. The people who cast the votes decide nothing. The people who count the votes decide everything.

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

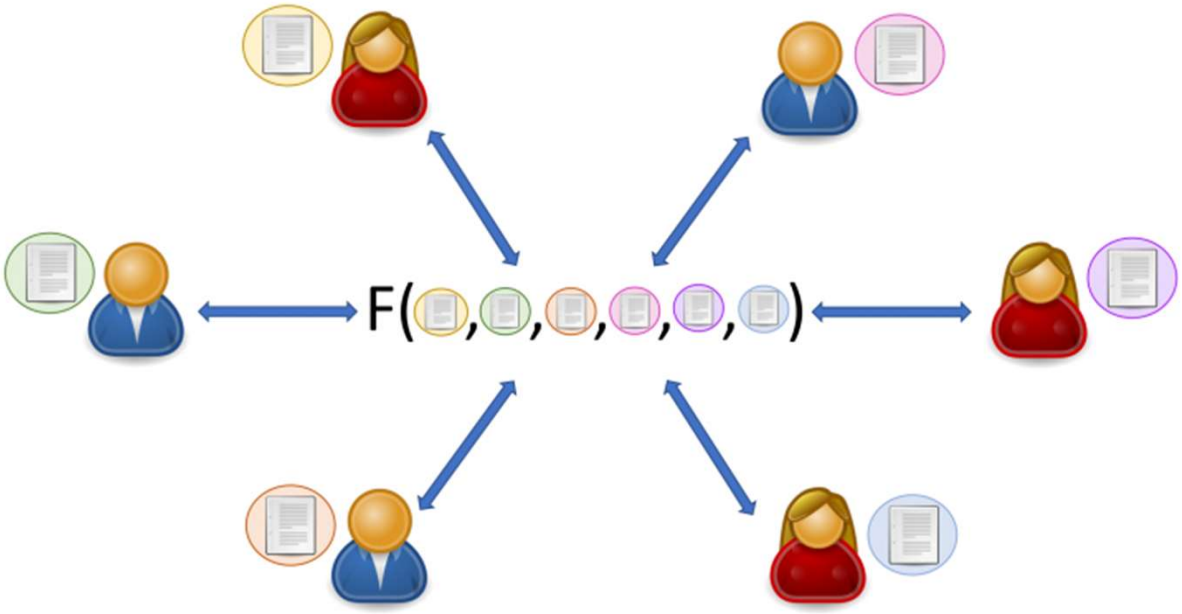
- Electronic Elections

… 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

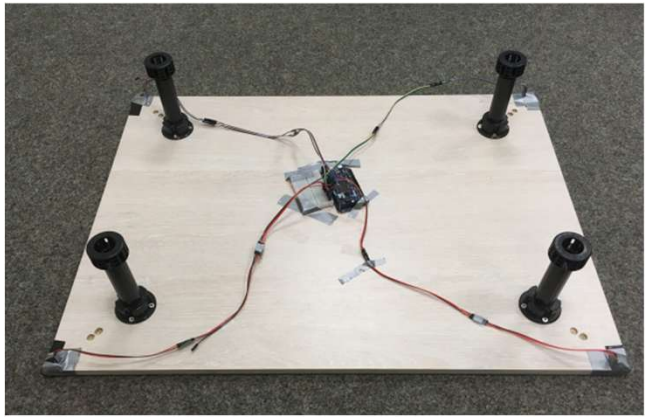
- The counting process is time-consuming.
- There are significant infrastructure expenses.
  - Ballots
  - Voting locations
  - Payment for trusted third parties
- Implementing intricate counting functions is challenging.
  - Solutions tend to raise costs
  - Elections that involve multiple rounds
-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

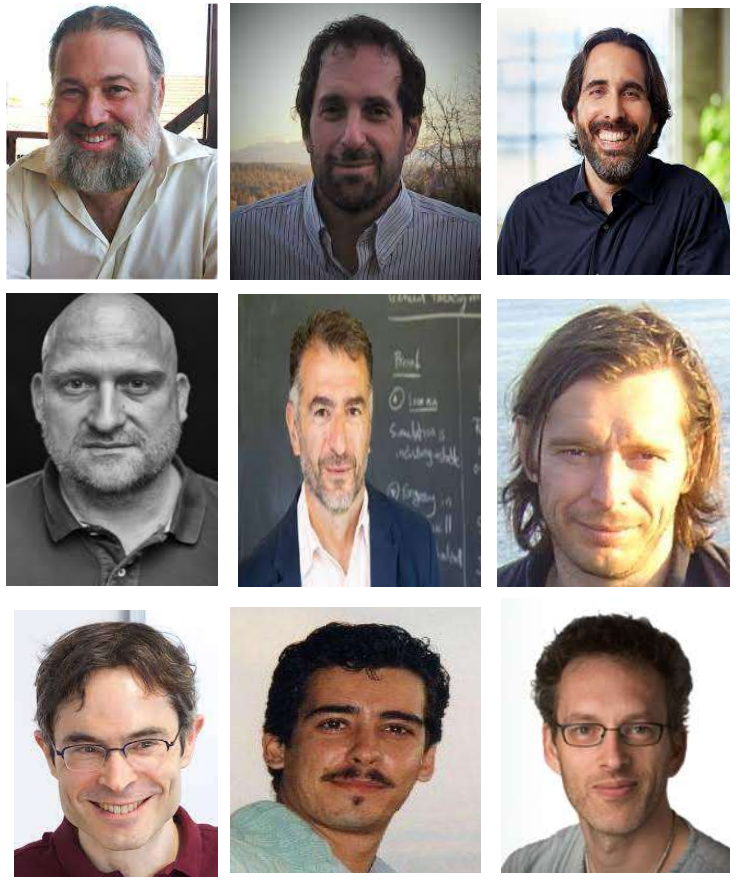
- In reality
  - Replace the ballot box with a computer
  - Input the voter choice
  - Electronic counting
  - No secrecy whatsoever!
- Voting with an untrusted intermediary
  - Malicious software
  - Programming errors
  - Targeted attacks
  - Interface problems
- No verifiability
- Open source: Necessary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

First generation electronic voting

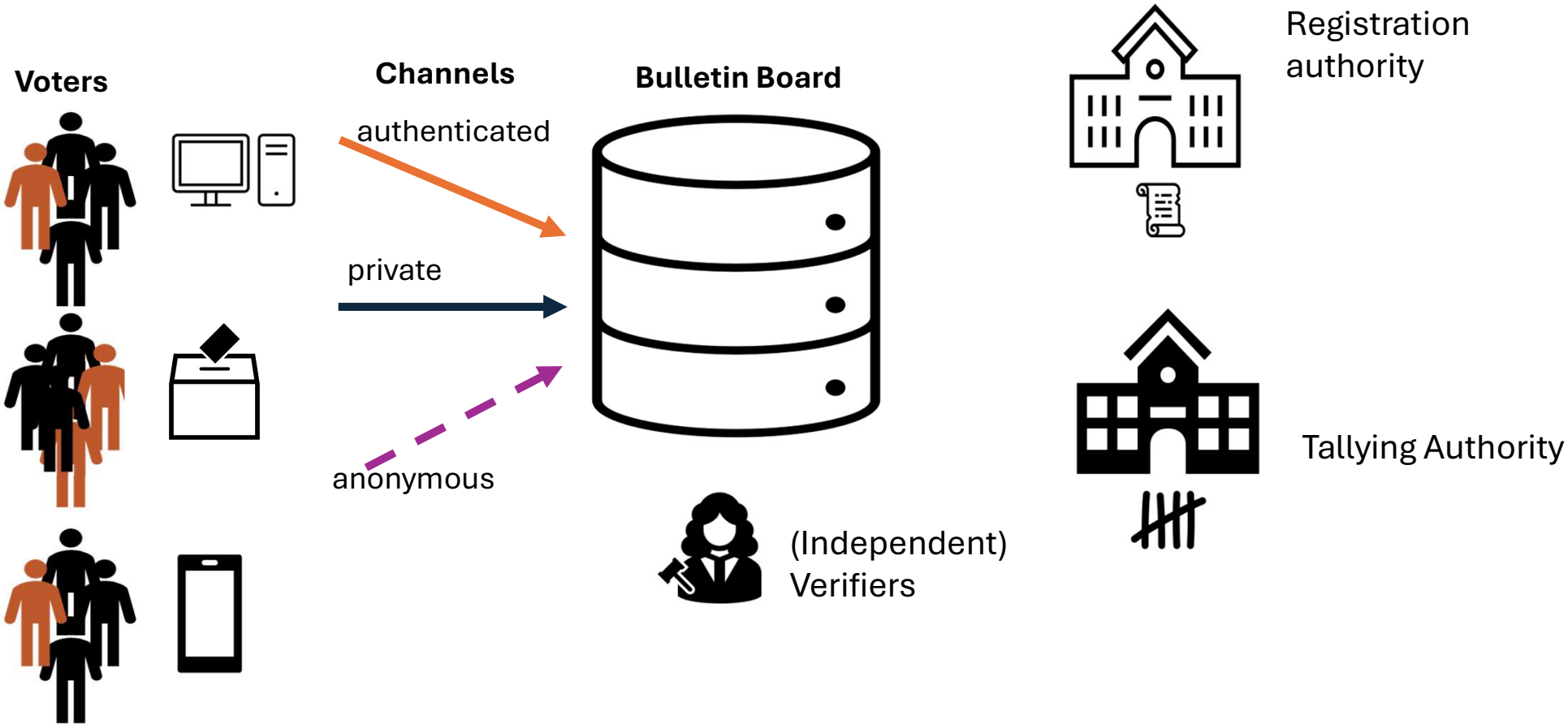
- Software independence (Rivest)
  - *A voting system is software-independent if an undetected change or error in its software cannot cause an undetectable change or error in an election outcome.*
- **Solution 1^{η}:**
  - Voter Verifiable Paper Trail (VVPAT) + Risk Limiting Audits (RLA)
- **Solution 2^{η}:**
  - 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
  - $sk \overset{\$}{\leftarrow} \mathbb{Z}_q, pk = g^{sk}$
  - pk belongs to the tallying authority
  - $Enc_{pk}(m) = (g^r, m \cdot pk^r), r \overset{\$}{\leftarrow} \mathbb{Z}_q, m \in \mathbb{G}$
  - $Dec_{sk}(c) = c_2 \cdot c_1^{-\mathrm{sk}} = \mathrm{m}$
- Exponential ElGamal
  - $Enc_{pk}(m) = (g^r, g^m \cdot pk^r), r \overset{\$}{\leftarrow} \mathbb{Z}_q, m \in \mathbb{Z}_q$
  - $Dec_{sk}(c) = c_2 \cdot c_1^{-\mathrm{sk}} = \mathrm{g^m}$
  - 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

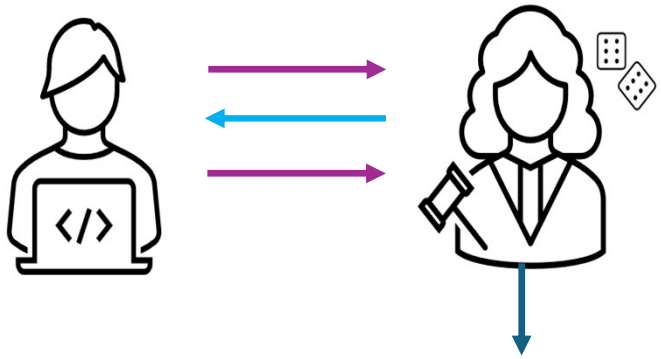
Basic Idea:

- 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!

Schnorr's Protocol

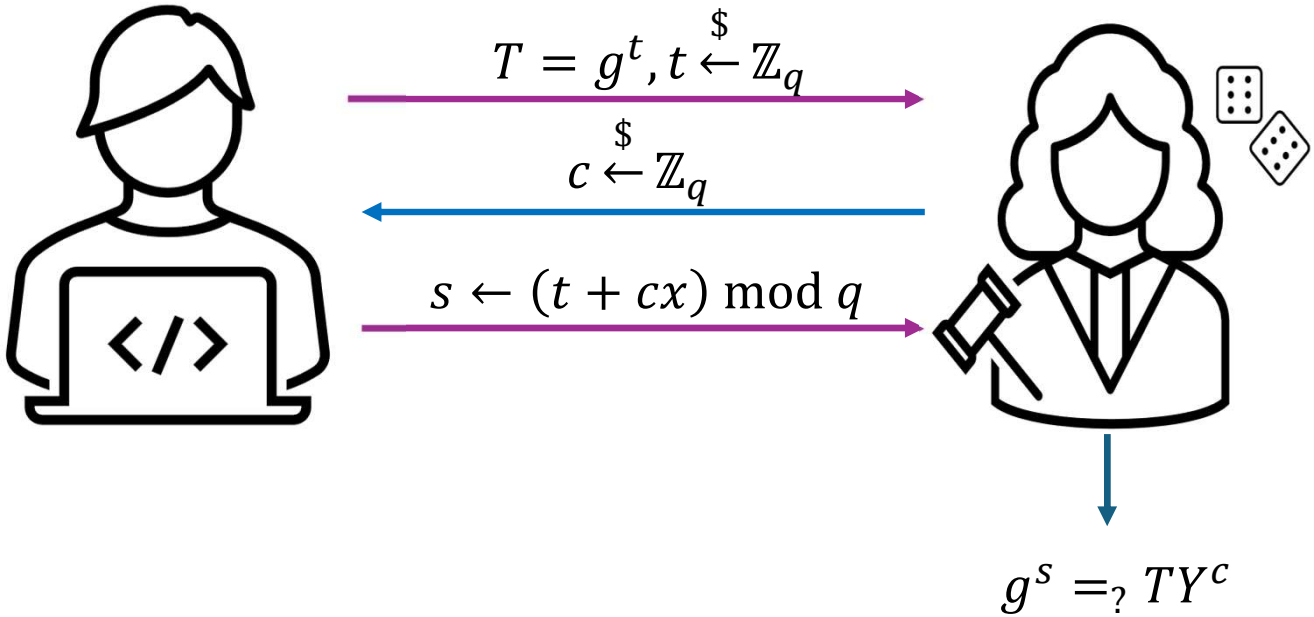


- Proof of Knowledge of a Discrete Logarithm
- $PoK\{x: g^x = Y : Y, g \in \mathbb{G}\}$
- Public Input
  - $\mathbb{G}$ is a cyclic group of prime order q generated by g
  - A group element $Y \in \mathbb{G}$
- Witness
  - $x \in \mathbb{Z}_q$

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: $x \in \mathbb{Z}_q : Y = g^x$

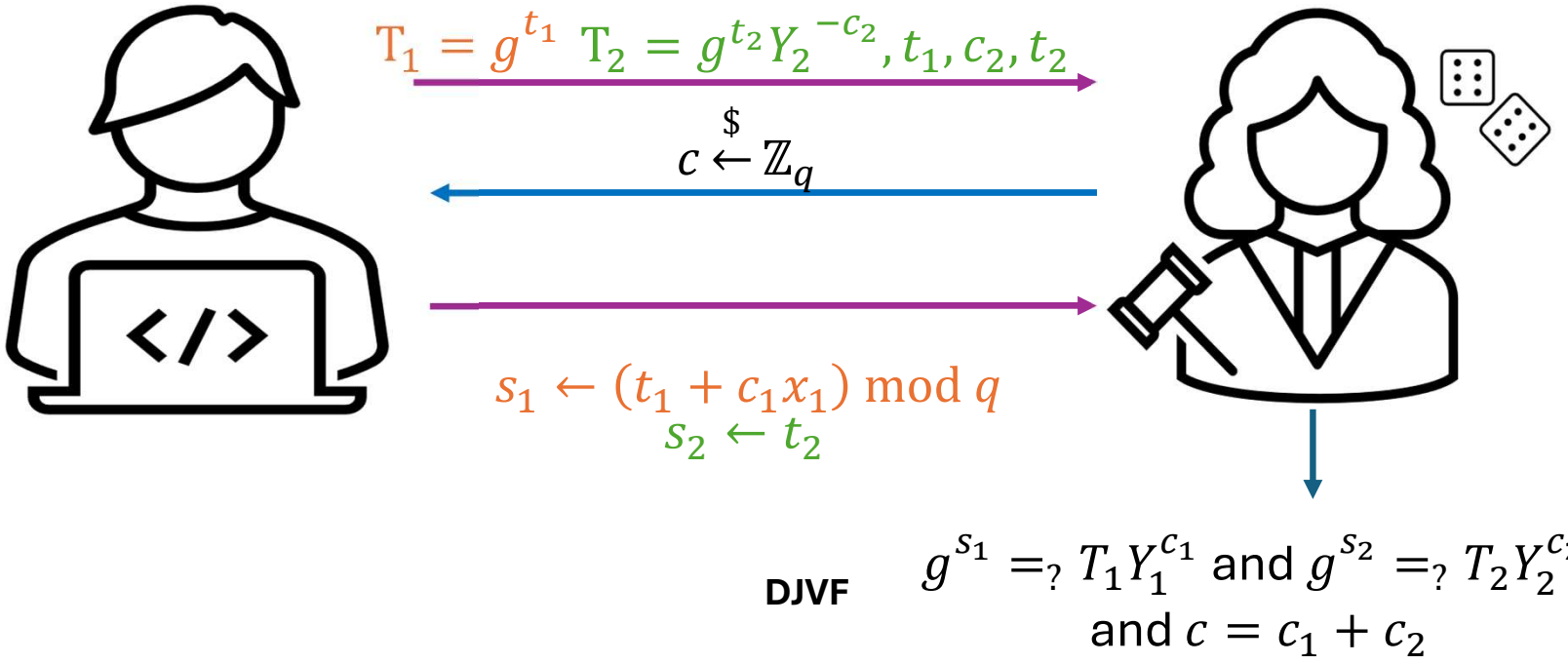
$\boldsymbol{DLPRV}(x, g, Y)$

- 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
- $PoK\{(\boldsymbol{x_1}, x_2): g^{\boldsymbol{x_1}} = Y_1 \textbf{ OR } g^{x_2} = Y_2,\ Y_1, Y_2, g \in \mathbb{G}\}$



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*

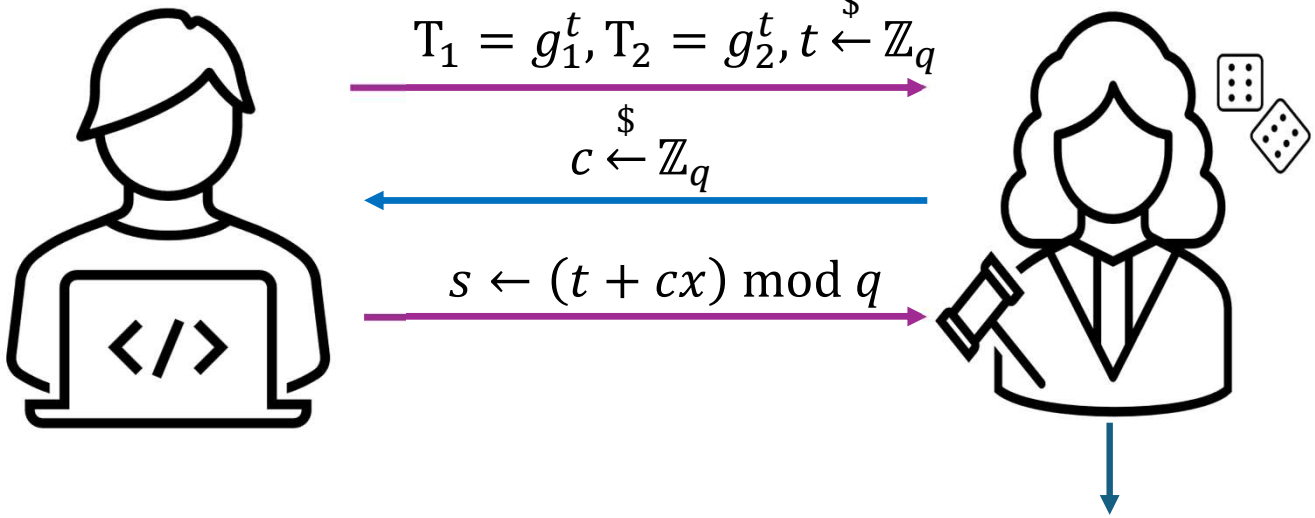
Pitfalls of the Fiat-Shamir Heuristic: An Example

- $\boldsymbol{DLPRV}(x, g, Y)$ proves knowledge of DLOG for a particular $Y \in \mathbb{G}$ which given as input to the prover
- If Y could be selected adaptively (after the proof):
  - Select $T \xleftarrow{\$} \mathbb{G}$
  - Compute $c \xleftarrow{\$} H(T)$
  - Select $s \xleftarrow{\$} \mathbb{Z}_q$
- The tuple (T, c, s) is a proof of knowledge for $Y = (g^{-s}T)^{-\frac{1}{c}}$ for which the DLOG is not known!
- Indeed:
  - $\boldsymbol{DLVF}(g, Y, \pi)$ checks in reality if $g^s Y^{-c} = T$ which holds by construction
  - $g^s Y^{-c} = g^s \left((g^{-s}T)^{-\frac{1}{c}}\right)^{-c} = T$

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
- $PoK\{x: g_1^x = Y_1, g_2^x = Y_2 : Y_1, Y_2, g_1, g_2 \in \mathbb{G}\}$



EQVF $g^s =_? T_1 Y_1^c$ and $g^s =_? T_2 Y_2^c$

Pitfalls of the Fiat-Shamir Heuristic (again)

$\boldsymbol{EQPRV}(x, g_1, Y_1, g_2, Y_2)$ proves knowledge and equality of DLOG for a **particular** $Y_1, Y_2 \in \mathbb{G}$ which is given as input to the prover

- If Y_1, g_2, Y_2 could be selected adaptively (after the proof):
  - Select $\alpha, \beta, \gamma, \delta \xleftarrow{\$} \mathbb{Z}_q$
  - Compute $T_1 \leftarrow g_1^{\alpha}, \mathrm{T}_2 \leftarrow g_1^{\beta}, g_2 \leftarrow g_1^{\gamma}, Y_2 \leftarrow g_1^{\delta}$
  - Compute $c \leftarrow H(T_1, T_2)$
  - Compute $s \leftarrow \frac{\beta + c\delta}{\gamma}$
  - Compute $x \leftarrow \frac{s-a}{c}$ and $Y_1 \leftarrow g^x$
  - $x \neq \log_{g_2} Y_2 \leftarrow \delta/\gamma$ but the proof verifies!
- Indeed, $\boldsymbol{EQVF}$ returns true:
  - $g_1^s Y_1^{-c} = g_1^{s - c\frac{s-a}{c}} = g_1^a = T_1$ and
  - $g_2^s Y_2^{-c} = g_1^{\gamma s} = g_1^{\gamma \frac{\beta + c\delta}{\gamma} - \delta c} = g_1^{\beta} = T_2$

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

Application to modern constructions

- FROZEN HEART (FoRging Of ZEro kNowledge proofs)
  - Girault's proof of knowledge protocol (Schnorr over a composite modulus)
  - Bulletproofs
  - PLONK

Takeaway
The Fiat-Shamir hash computation must include all public values from the zero-knowledge proof statement and all public values computed intermediately the proof (i.e., all random "commitment" values)

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 $c = H(g, g^r, g^s(g^r)^{-c})$

Enc+PoK for non-malleability

- If wFS is used, the ciphertext is still malleable, despite the existence of the proof!

- Given $\boldsymbol{c_1} = (R, S, c, s)$
- Create $\boldsymbol{c_2} = (Rg^u, SY^u, c, s + cu), u \xleftarrow{\$} \mathbb{Z}_q$
- The proof still verifies:
  - $g^{(s+cu)}(Rg^u)^{-c} = g^s R^{-c}$
  - which is valid from the original proof (c, s)
  - But the ciphertext has changed!

Enc+PoK with Strong FS implies NM-CPA

NM-CPA Game

$(pk, sk) \leftarrow KGen(1^{\lambda})$
$m_0, m_1 \leftarrow A(pk)$
$b \xleftarrow{\$} \{0,1\}$
$c^* \leftarrow Enc_{pk}(m_b)$
$\mathbf{c} \leftarrow A(pk, c^*)$ // parallel CCA
$for\ c_1 \in \mathbf{c}$:
  $if\ c_i = c^*\ return\ \perp$
  $else\ m_i \leftarrow Dec_{sk}(c_i)$
$b^* \leftarrow A(pk, c^*, \mathbf{m})$
$return\ b = b^*$

The contents of the vector $\boldsymbol{c}$ are created independently

Mihir Bellare and Amit Sahai, Non-malleable encryption: Equivalence between two notions, and an indistinguishability-based characterization, CRYPTO' 99

Breaking NM-CPA implies breaking IND-CPA

Hybrid 1:
Use the PoK Simulator when constructing c^*
If the adversary can distinguish the simulated proof then it can break ZK of PoK

Hybrid 2:
In order to construct $\mathbf{c}$, A makes random oracle queries for $\boldsymbol{DLPRV}$. The challenger uses the ZK extractor to retrieve the witness r for the proof and decrypt c_i without sk ($m_i = c_2/pk^{-r}$)
This is the IND-CPA Game

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

Plaintext equivalence test (**PET**)

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

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

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

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

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

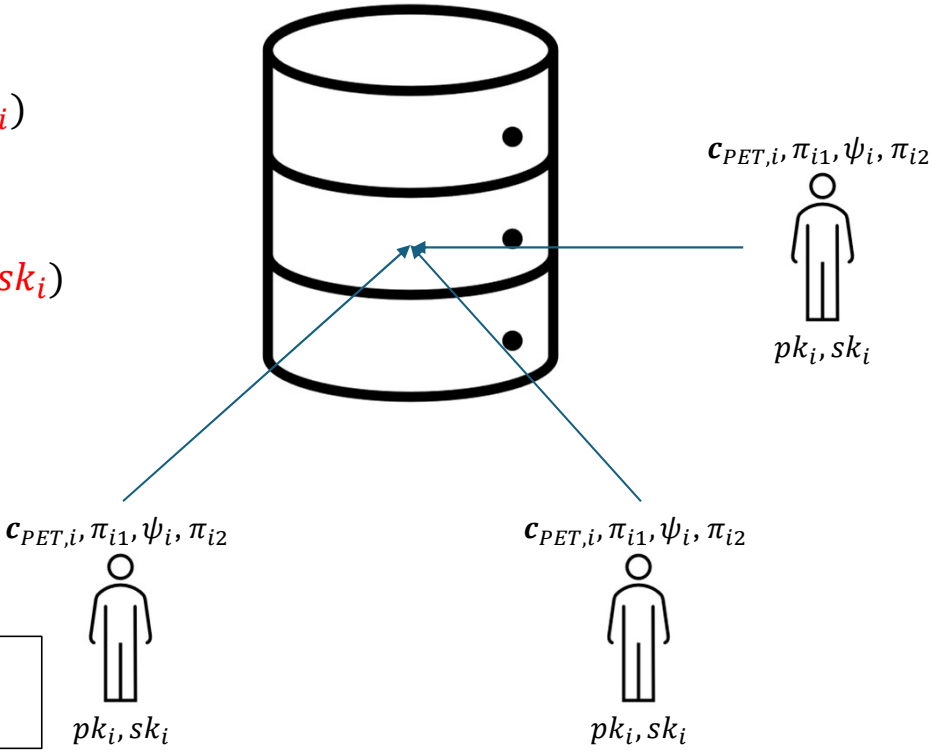
$$\psi_i = x^{sk_i} \quad \pi_{i2} = DLRPV(sk_i)$$

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

$$\rho =_? 1$$

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

Blind Signatures

- A set of algorithms $\Pi = (KGen, Sign, Vf)$
- $(sk, vk) \leftarrow KGen(1^{\lambda})$
- $\sigma \leftarrow \mathbf{Sign}\langle S(sk), U(m), vk\rangle$
  - **Sign** is a **protocol** and not an algorithm. It consists of:
    - $m' \leftarrow Blind(m, vk)$ initiated by the U
    - $\sigma' \leftarrow Sign(m', sk)$ initiated by the S - ***Sign*** is an algorithm
    - $\sigma \leftarrow Unblind(\sigma', vk)$ executed by the U
- Verification: $\{0,1\} \leftarrow Vf(m, \sigma, vk)$
- Correctness:
  - $Vf(m, \mathrm{Sign}\langle S(sk), U(m), vk\rangle, vk) = 1$: $(sk, vk, prms) \leftarrow KGen(1^{\lambda})$

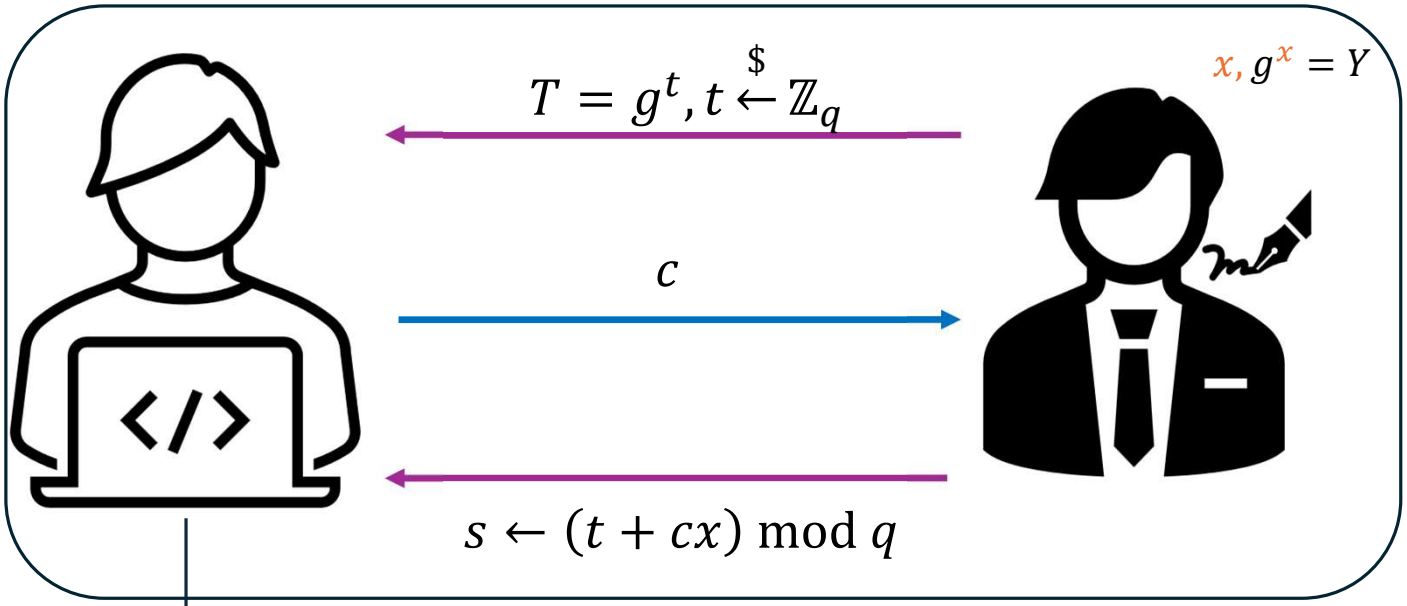
Security Properties

- Blindness
  - The adversary is the signer
  - Goal: Indistinguishability of signatures-signing sessions
- Unforgeability
  - The adversary is the user
  - A blind signature is a forgery (created by the user not the signer)
  - One-more unforgeability
    - The user may not create more signatures than signing sessions

RSA Blind Signatures

- Key generation
  - Like in plain RSA. Finally: $(sk, vk) = (d, (e, n))$
- Signing:
  - $Blind(m, vk) \rightarrow H(m) \cdot r^e \bmod n, r \leftarrow Z_n^*$
  - $Sign(m', sk) \rightarrow m'^d \bmod n \rightarrow \left(H(m)^d r\right) \bmod n$
  - $UnBlind(\sigma', vk) \rightarrow \sigma' r^{-1} \bmod n \rightarrow H(m)^d \bmod n$
- Verification:
  - The unblinded signature is a simple RSA signature

Schnorr Blind Signatures



α, β are blinding factors to alter (c, s) from (c', s')

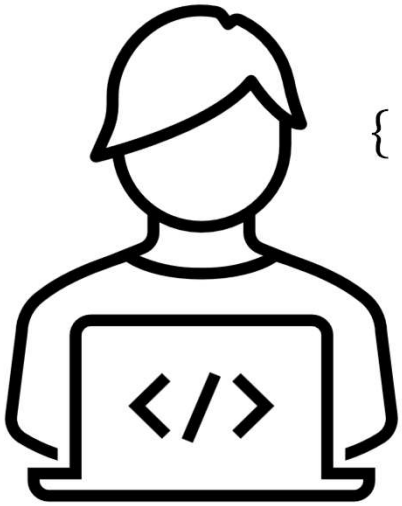
$g^{s'} \cdot Y^{c'} = g^{s+a} \cdot Y^{\beta+c'} = Tg^{\alpha}Y^{\beta} = T'$

Group-Ring Signatures

- A set of algorithms $\Pi = (KGen, Sign, Vf)$
- $(sk, vk) \leftarrow KGen(1^{\lambda})$
- $\sigma \leftarrow \mathbf{Sign}(sk, m, \boldsymbol{R})$
  - **R is a set of public keys**
  - sk corresponds to a $pk \in R$
- Verification: $\{0,1\} \leftarrow Vf(m, \sigma, \boldsymbol{R})$
- Basic idea:
  - Hide the signer inside a group of peers
  - Ring signatures: The group is **ad/hoc**
  - Group signatures: There is a group manager who might manages the group and might trace the signer

Ring Signatures via OR proofs

$R = \{Y_1. Y_2, \dots, Y_n\}$ $\quad PoK\{\boldsymbol{x}_k : g^{x_k} \in R : g^{x_k} = Y_1 \textbf{ OR } g^{x_k} = Y_2, \textbf{OR } \dots, g^{x_k} = Y_n\}$



$$T_\kappa = g^{t_\kappa},$$

$$\{T_i = g^{t_i} Y_i^{-c_i}, t_i, c_i \xleftarrow{\$} \mathbb{Z}_q\} \quad i \in [n]/\{k\}$$

$$c \leftarrow H(R, \boldsymbol{m}, \{T_i\}_{i=1}^{n})$$

$$c_k \leftarrow c - \sum c_i$$

$$s_k \leftarrow (t_k + c_k x_k) \bmod q$$

$$s_i \leftarrow t_i, i \in [n]/\{k\}$$

$$\forall i \in [n]\ g^{s_i} =_? T_i Y_i^{c_i} \quad \text{και } c =_? \sum_i c_i \bmod q$$

Linkable Ring Signatures

- Two signatures by the same signer can be identified as such, but the signer remains anonymous.
- Enabled by including a linking tag in the signature
  - A function of the secret key
- Eliminate double voting
- Enable revoting

Liu, J.K., Wei, V.K., Wong, D.S. (2004). Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups. In: Wang, H., Pieprzyk, J., Varadharajan, V. (eds) Information Security and Privacy. ACISP 2004. Lecture Notes in Computer Science, vol 3108. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-27800-9_28

Linkable Ring Signatures

$R = \{Y_1, Y_2, \dots, Y_n\}$

$s_{\pi+1} \overset{\$}{\leftarrow} \mathbb{Z}_q$

$c_{\pi+2} \leftarrow H_q(R, m, tag, g^{s_{\pi+1}} Y_i^{s_{\pi+1}}, h^{s_{\pi+1}} tag^{s_{\pi+1}})$

$c_{\pi+1}$

$c_{\pi+2}$

...

$t \overset{\$}{\leftarrow} \mathbb{Z}_q$

$c_{\pi+1} \leftarrow H_q(R, m, tag, g^t, h^t)$

x_π, Y_π

$h \leftarrow H_{\mathbb{G}}(R)$

$tag \leftarrow h^{x_\pi}$

$s_i \overset{\$}{\leftarrow} \mathbb{Z}_q$

$c_{i+1} \leftarrow H_q(R, m, tag, g^{s_i} Y_i^{s_i}, h^{s_i} tag^{s_i})$

$s_\pi \leftarrow t - c_\pi x_\pi$

$\sigma = (c_1, s_1, \cdots, s_n, tag)$

c_π

c_{i+1}

$s_{\pi-1} \overset{\$}{\leftarrow} \mathbb{Z}_q$

$c_\pi \leftarrow H_q(R, m, tag, g^{s_{\pi-1}} Y_i^{s_{\pi-1}}, h^{s_{\pi-1}} tag^{s_{\pi-1}})$

Public Verification:

$\sigma = (c_1, s_1, \cdots, s_n, tag)$

Recompute $h \leftarrow H_{\mathbb{G}}(R)$

Compute for $i = 1 \dots n$

$c_{i+1} \leftarrow H_q(R, m, tag, g^{s_i} Y_i^{s_i}, h^{s_i} tag^{s_i})$

Check if

$c_1 = c_{n+1}$

Security Properties

- Unforgeability
  - No one can create a signature without knowing a secret key
- Anonymity
  - No-one can identify which ring member is the actual signer
- Linkability
  - Two signatures from the same signer are can be linked together
- Non-Slanderability
  - Given a signature no one can create a signature that links to it except for the original signer

Designated-Verifier Signatures

- Restrict the public verifiability of digital signatures
- Only a specific entity, *designated during signature creation,* can be sure about the signer of a message
  - Designation – inclusion of its public key in the signature algorithm
- Main idea:
  - Create an OR-proof of the statement:
    - **I know the signer's private key OR the designated verifier's private key**
  - The designated verifier can simulate proofs
    - Extra functionality **Simulate**
  - Other entities cannot be sure of the actual signer

Markus Jakobsson, Kazue Sako, and Russell Impagliazzo. "Designated Verifier Proofs and Their Applications". In: *EUROCRYPT '96*

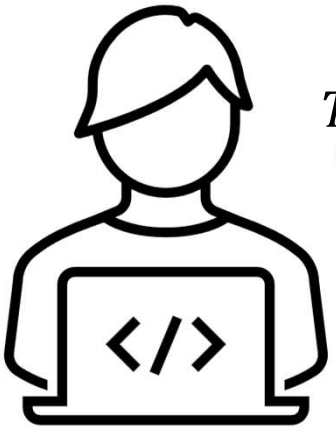
Designated-Verifier Signatures

- Strong DV Signatures
  - The private key of the DV is required for verification
- Use in e-voting:
  - Deniability for coercion-resistant and receipt-free schemes
  - The voters can simulate proofs that they followed the instructions of the coercer
- Security Properties
  - Unforgeability
    - Only the DV can ‘forge’ signatures
  - Non-Transferability
    - The public cannot tell if a signature originates from the signer or the DV

Designated-Verifier Schnorr

Proof

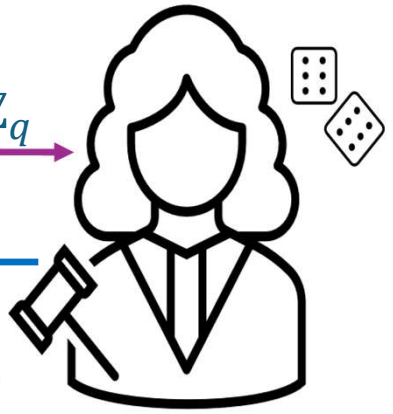
$$PoK_V\{x : g^x = Y : Y, Y_V, g \in \mathbb{G}\}$$



$T = g^t, t \xleftarrow{\$} \mathbb{Z}_q \quad Z = g^w Y_V^r, w, r \xleftarrow{\$} \mathbb{Z}_q$

$c \leftarrow H(Y, Y_V, T, Z)$

$s \leftarrow (t + (c + w)x) \bmod q$



(w, r, T, s)

$Z = g^w Y_V^r$

$c \leftarrow H(Y, Y_V, T, Z)$

$g^s =_? TY^{c+w}$

Simulation

$$T \leftarrow g^s Y^{-\beta}, \qquad Z \leftarrow g^a, \qquad a, \beta, s \xleftarrow{\$} \mathbb{Z}_q$$

$$c \leftarrow H(Y, Y_V, T, Z)$$
$$r \leftarrow (a - w) \cdot x_v^{-1}$$
$$w \leftarrow (\beta - c)$$

Return (w, r, T, s)