LMS Computer Science Colloquium 2019

Security Models and Designs from E-Voting to Blockchain

Dr David Galindo

Senior Lecturer in Computer Security, University of Birmingham

Head of Cryptography, Fetch.AI, Cambridge

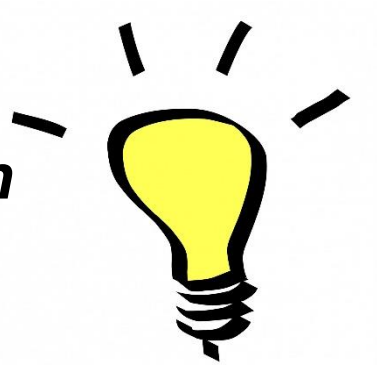
https://www.dgalindo.es

Outline

- Computational cryptography approaches to **defining and measuring security**
- Scenarios: electronic **voting** and **blockchain**
- Canonical **designs**

based on joint work with:
Véronique Cortier, David Bernhard, Sandra Guasch, Bogdan Warinschi, Olivier Pereira, Alex Escala

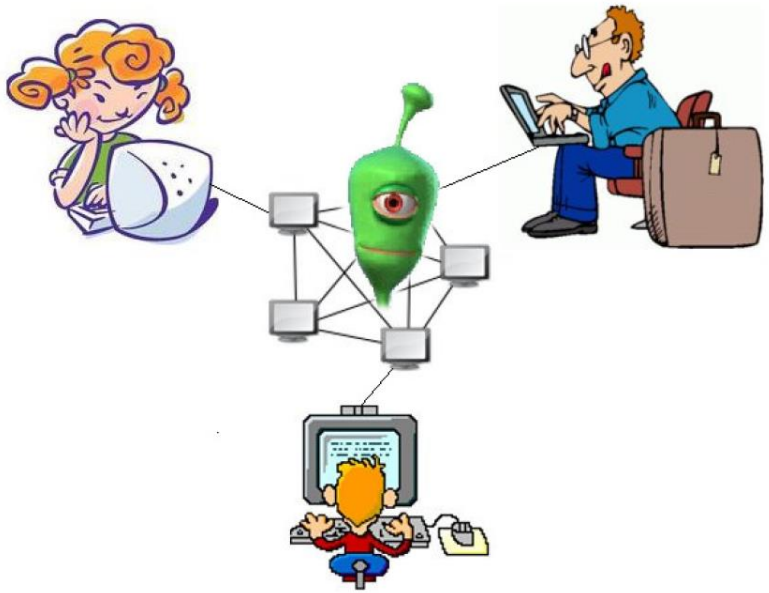
***Models and designs are* relative, *dependent on research community* ethos *and* incentives**



Cryptographic protocols

= is an abstract or concrete protocol that performs a security-related function and applies cryptographic methods, often as sequences of cryptographic primitives. A protocol describes how the algorithms should be used (Wikipedia)

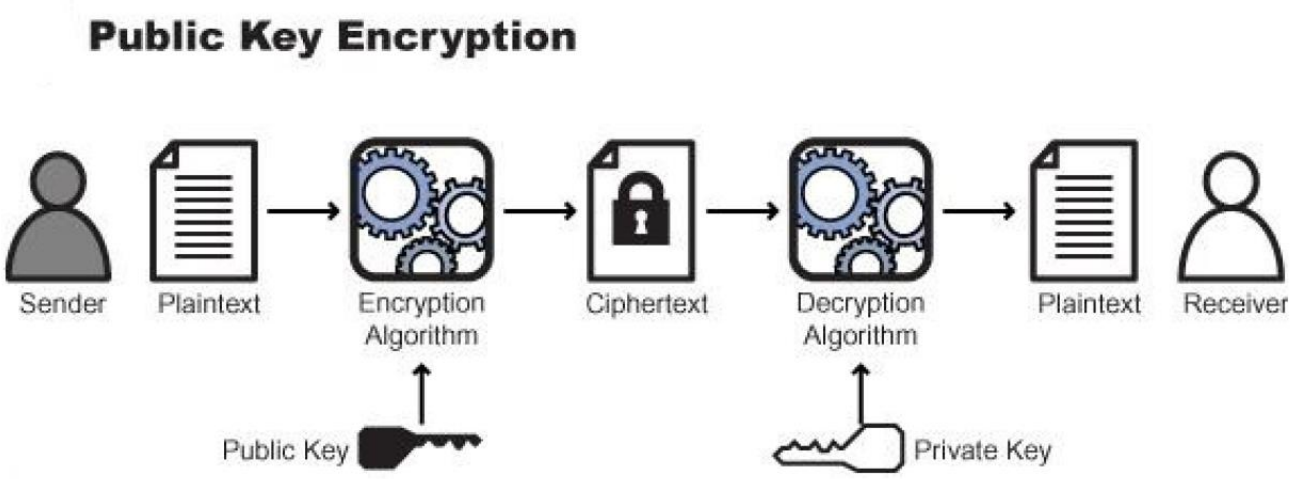
Adversarial capabilities



Adversaries might be able to:

- **read** exchanged messages
- **intercept** communications
- **build and send** messages
- **participate** in the protocols

Public Key Encryption

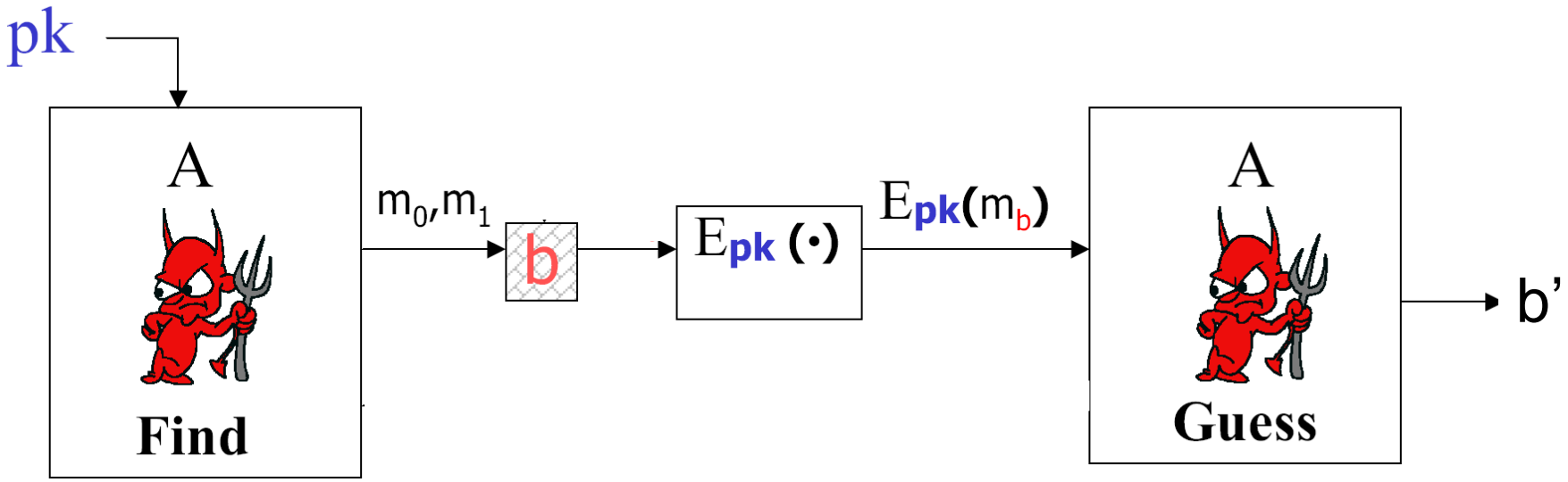


Public Key Encryption

A public key encryption scheme consists of the following algorithms $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$:

- $\mathsf{KG}(\mathsf{pms}(\lambda))$ on input **global parameters** pms outputs pair of encryption/decryption keys (PK, SK)
- $\mathsf{Enc}(PK, m; r)$ on inputs a public key PK, plaintext m outputs a ciphertext C (eventually local randomness r)
- $\mathsf{Dec}(SK, C)$ on inputs a decryption key SK and a ciphertext C outputs a plaintext m

Indistinguishability of encryptions



Prob[b'= b] ?

Indistinguishability of encryptions (IND-CPA)

IND-CPA

Init On input PK from $(PK, SK) \leftarrow \mathrm{KG}(\mathrm{pms})$, adversary $\mathcal{A}$ outputs m_0, m_1 and $C_\beta \leftarrow \mathrm{Enc}(PK, m_\beta)$ is computed for $\beta \xleftarrow{\$} \{0,1\}$

Guess $\mathcal{A}(PK, C_\beta)$ outputs a bit β' and wins if $\beta' = \beta$

$\mathrm{AdvIND\text{-}CPA} = |\Pr[\beta = \beta'] - 1/2|$ shall be negligible for any PPT adversary $\mathcal{A}$

Diffie-Hellman Groups

- Let q be a prime number
- Let $\mathbb{G}$ be a commutative **cyclic group** wrt to a **product operation** $(g_1, g_2) \stackrel{\cdot}{\mapsto} g_1 \cdot g_2$ namely

  $$\mathbb{G} = \{1_{\mathbb{G}}, g, g^2, \ldots, g^{q-1}\}, \text{ where } g^q = g^0 = 1_{\mathbb{G}}$$

  for $g \in \mathbb{G}$ (g is called **generator**). We write $\mathbb{G} = \langle g \rangle$
- Any $h \in \mathbb{G}$ can be uniquely written as $h = g^x$ with $0 \leq x < q$; equivalently $x \in \mathbb{Z}_q$
- The integer x is called the **discrete logarithm** of h to the base g, and denoted $\log_g h$ or $\mathrm{DLog}_g(h)$
- The order of a group $\mathbb{G}$ is its number of elements, denoted $|\mathbb{G}|$ or $\mathrm{ord}(\mathbb{G})$

DH Problems Family

- Discrete logarithm (**DL**) problem:
  - Given $\mathbb{G} = \langle g \rangle$ and $g, h \in \mathbb{G}$ with $h = g^x$ for *unknown and random* $x \in \mathbb{Z}_q$ **compute** $x = \mathrm{DLog}_g(h)$
- Computational Diffie-Hellman (**CDH**) problem:
  - Given $g, h_1 = g^{x_1}, h_2 = g^{x_2} \in \mathbb{G}$ for *unknown and random* $x_1, x_2 \in \mathbb{Z}_q$ **compute** $g^{x_1 x_2}$
- Decision Diffie-Hellman (**DDH**) problem:
  - For *unknown and random* $x, y, z \in \mathbb{Z}_q$, **distinguish** the tuple (g, g^x, g^y, g^z) from (g, g^x, g^y, g^{xy})

DH Group Instantiations

NIST P-224
Curve1174
Curve25519
BN(2,254)
brainpoolP256t1
ANSSI FRP256v1
NIST P-256
secp256k1

Curve383187
brainpoolP384t1
NIST P-384
Curve41417
Ed448-Goldilocks

DH Group Instantiations (NIST)

Date	Minimum of Strength	Symmetric Algorithms	Factoring Modulus	Discrete Logarithm Key	Discrete Logarithm Group	Elliptic Curve	Hash (A)	Hash (B)
(Legacy)	80	2TDEA*	1024	160	1024	160	SHA-1**	
2016 - 2030	112	3TDEA	2048	224	2048	224	SHA-224 SHA-512/224 SHA3-224	
2016 - 2030 & beyond	128	AES-128	3072	256	3072	256	SHA-256 SHA-512/256 SHA3-256	SHA-1
2016 - 2030 & beyond	192	AES-192	7680	384	7680	384	SHA-384 SHA3-384	SHA-224 SHA-512/224
2016 - 2030 & beyond	256	AES-256	15360	512	15360	512	SHA-512 SHA3-512	SHA-256 SHA-512/256 SHA-384 SHA-512 SHA3-512

https://www.keylength.com/

ElGamal encryption (1985)

- Stp(λ) choose $\mathbb{G} = \langle g \rangle$ to be a q prime-order group with $\lambda = \lceil \log q/2 \rceil$. Set pms $\leftarrow (\mathbb{G}, \mathbb{Z}_q)$

The following encryption scheme is IND-CPA secure under the DDH assumption:

- KG(pms) choose $g_1, g_2 = g_1^a \in \mathbb{G}$, and set $PK = (g_1, g_2)$ and $SK = a \xleftarrow{\$} \mathbb{Z}_q$
- Enc(PK, m) to encrypt a "not too large" $m \in \mathbb{Z}_\tau^+$, choose $r \xleftarrow{\$} \mathbb{Z}_q$ and output $C = (g_1^r, g_2^r \cdot g^m)$
- Dec(SK, C) given $C = (c, d)$ output $d/(c^a)$ and search message space for m

ElGamal is malleable

Given:

- $\text{Enc}(PK, m) = (c_1, c_2) = (g^r, h^r \cdot g^m)$
- $\text{Enc}(PK, m') = (c_1', c_2') = (g^{r'}, h^{r'} \cdot g^{m'})$
- $\text{Enc}(PK, m) \otimes \text{Enc}(PK, m') = (c_1 \cdot c_1', c_2 \cdot c_2')$
- Then an encryption of $m + m'$ can be obtained by "multiplying" the previous ciphertexts:

  $\text{Enc}(PK, m) \otimes \text{Enc}(PK, m') = (g^r \cdot g^{r'}, h^r \cdot h^{r'} \cdot g^m \cdot g^{m'}) =$
  $(g^{r+r'}, h^{r+r'} \cdot g^{m+m'}) = (g^s, h^s \cdot g^{m+m'}) = \text{Enc}(PK, m + m')$
  where $s := r + r' \mod q$

Non-malleability

Init On input PK from $(PK, SK) \leftarrow \mathrm{KG}(\mathrm{pms})$, adversary $\mathcal{A}$ outputs m_0, m_1 and $C_\beta \leftarrow \mathrm{Enc}(PK, m_\beta)$ is computed for $\beta \xleftarrow{\$} \{0,1\}$

Find $\mathcal{A}$ submits a ciphertext vector $\boldsymbol{c} = (c_i)_i$ and obtains $(m_i \leftarrow \mathrm{Dec}(SK, c_i))_i$, where $c_i \neq C_\beta$

Guess $\mathcal{A}(PK, C_\beta, (m_i)_i)$ outputs a bit β' and wins if $\beta' = \beta$

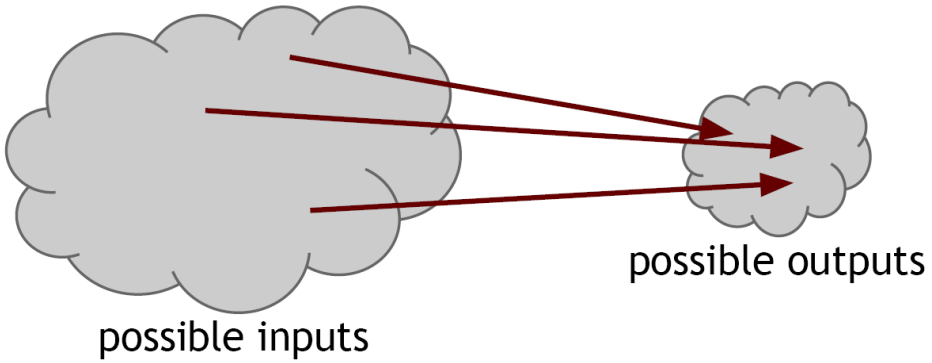
$\mathrm{AdvNM} = |\Pr[\beta = \beta'] - 1/2|$ shall be negligible for any PPT adversary $\mathcal{A}$

Hash functions

A hash function *H*:

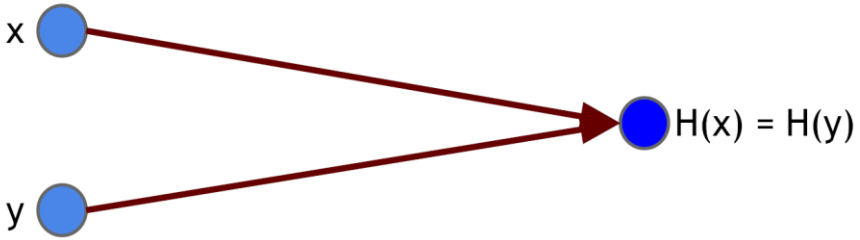
- takes any string as input
- fixed-size output (typically 256 bits)
- efficiently computable
- collision-free

Collisions do exist …



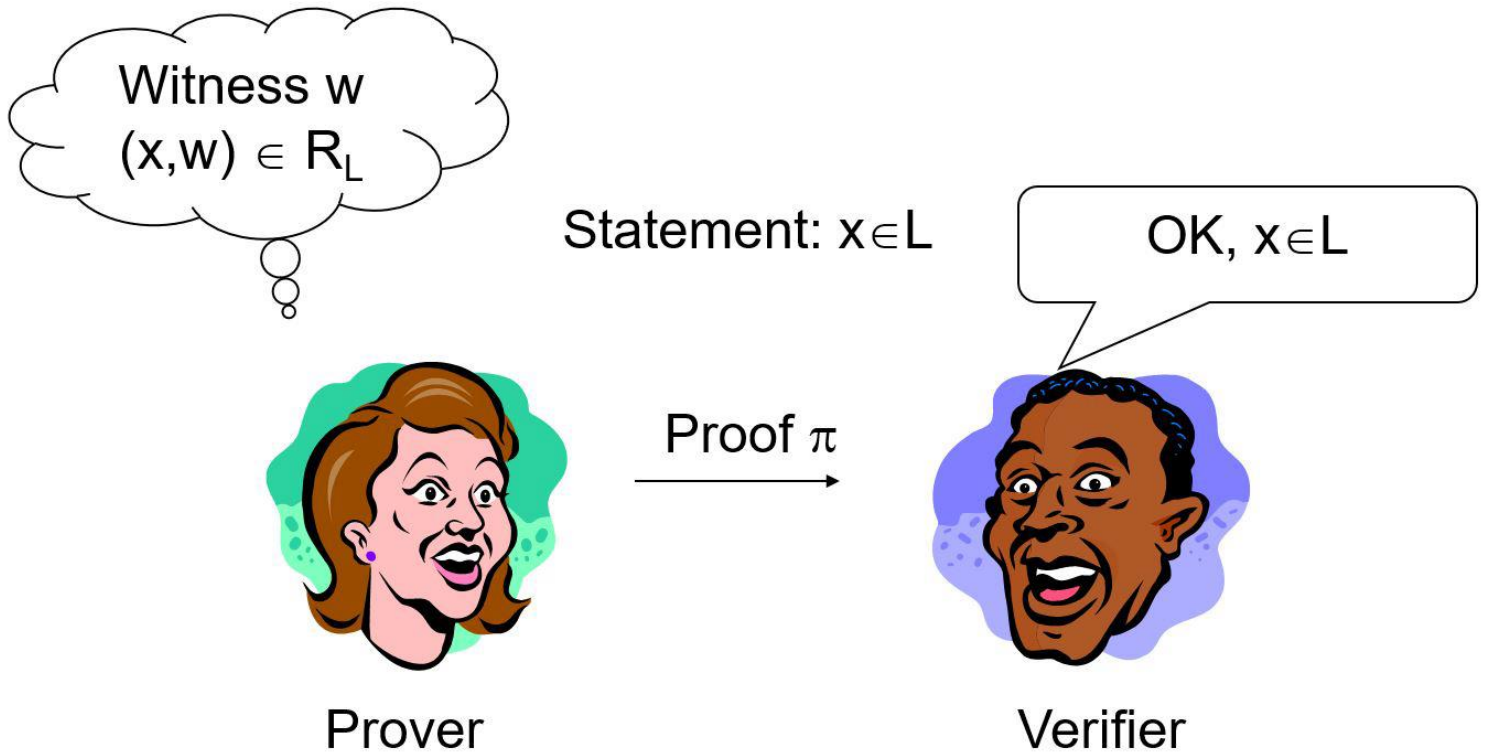
Nobody can find x and y such that

x != y and H(x)=H(y)



Proof systems

Non-interactive proofs



Zero Knowledge Proofs (Properties)

- **Completeness**: Given an honest prover and an honest verifier the protocol succeeds (with overwhelming probability)
- **Soundness:** if the statement is false no cheating prover can convince the honest verifier that is true (except with negligible probability)
- **Zero Knowledge:** an honest prover executing the protocol does not release any information about its secret witness other than that the particular assertion is true

Simulator

Extractor

ZKP Equality of Discrete Logarithms

Language $\mathcal{L}_{\text{EqDl}} = \{(g_1, g_2, X_1, X_2) \mid X_1 = g_1^w;\ X_2 = g_2^w\}$, namely $\log_g X_1 = \log_h X_2$.

The value $w \in \mathbb{Z}_q$ is called **witness**.

The **Equality of Discrete-Logarithms proof system** $\text{EqDl}(g, h, X_1, X_2) = (\text{PrEq}, \text{VerifyEq})$ works as follows:

- $\text{PrEq}(g, h, X_1, X_2, x)$ outputs a proof $\pi^{\text{eq}} = (c, s)$

  $R_1 = g^r$ and $R_2 = h^r$ for $r \xleftarrow{R} \mathbb{Z}_q$
  $c := H(X_1, X_2, R_1, R_2)$
  $s = r - x \cdot c \mod q$

- $\text{VerifyEq}(g, h, X_1, X_2, \pi^{\text{eq}} = (c, s))$ outputs `true` or `false`

  computes $R_1 := g^s \cdot X_1^c$
  computes $R_2 := h^s \cdot X_2^c$
  returns $c \stackrel{?}{=} H(X_1, X_2, R_1, R_2)$

where $H : \{0,1\}^* \to \mathbb{Z}_q$ is a hash function

Non-Malleable ElGamal

- $\mathsf{Stp}(\lambda)$ choose $\mathbb{G} = \langle g \rangle$ to be a q prime-order group with $\lambda = \lceil \log q/2 \rceil$. Let $H : \{0,1\}^\star \to \mathbb{Z}_q$. Set $\mathsf{pms} \leftarrow (\mathbb{G}, \mathbb{Z}_q, H)$

The following encryption scheme is **non-malleable** under the DDH assumption:

- $\mathsf{KG}(\mathsf{pms})$ choose $g_1, g_2 = g_1^a \in \mathbb{G}$, and set $PK = (g_1, g_2)$ and $SK = a \xleftarrow{\$} \mathbb{Z}_q$
- $\mathsf{Enc}(PK, m)$ to encrypt a "not too large" $m \in \mathbb{Z}_\tau$, choose $r \xleftarrow{\$} \mathbb{Z}_q$, compute $C = (g_1^r, g_2^r \cdot g^m)$ and $\pi = \mathsf{ZKP}(r)$. Output (C, π).
- $\mathsf{Dec}(SK, (C, \pi))$ given $C = (c, d)$, reject if proof π not correct. Otherwise, output $d/(c^a)$ and search message space for m

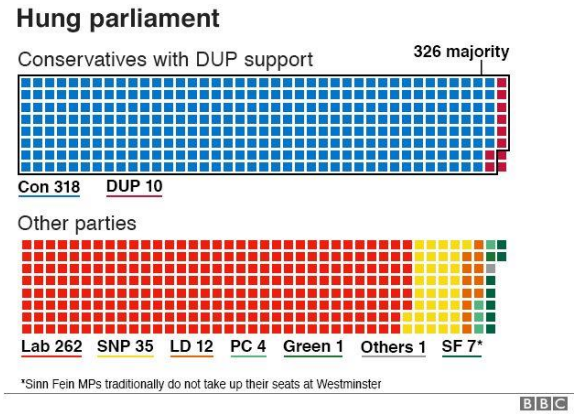
Electronic voting

Elections


ELECTORAL REGISTER


POLLING
STATION


BALLOT
BOX


Hung parliament
Conservatives with DUP support
326 majority
Con 318
DUP 10
Other parties
Lab 262
SNP 35
LD 12
PC 4
Green 1
Others 1
SF 7*
*Sinn Fein MPs traditionally do not take up their seats at Westminster
BBC

Remarks

- Elections are **centralized** yet **distributed** systems
- Centralized: Register, Tally
- Distributed: Polling, Voting

Online Voting

Voter Choices

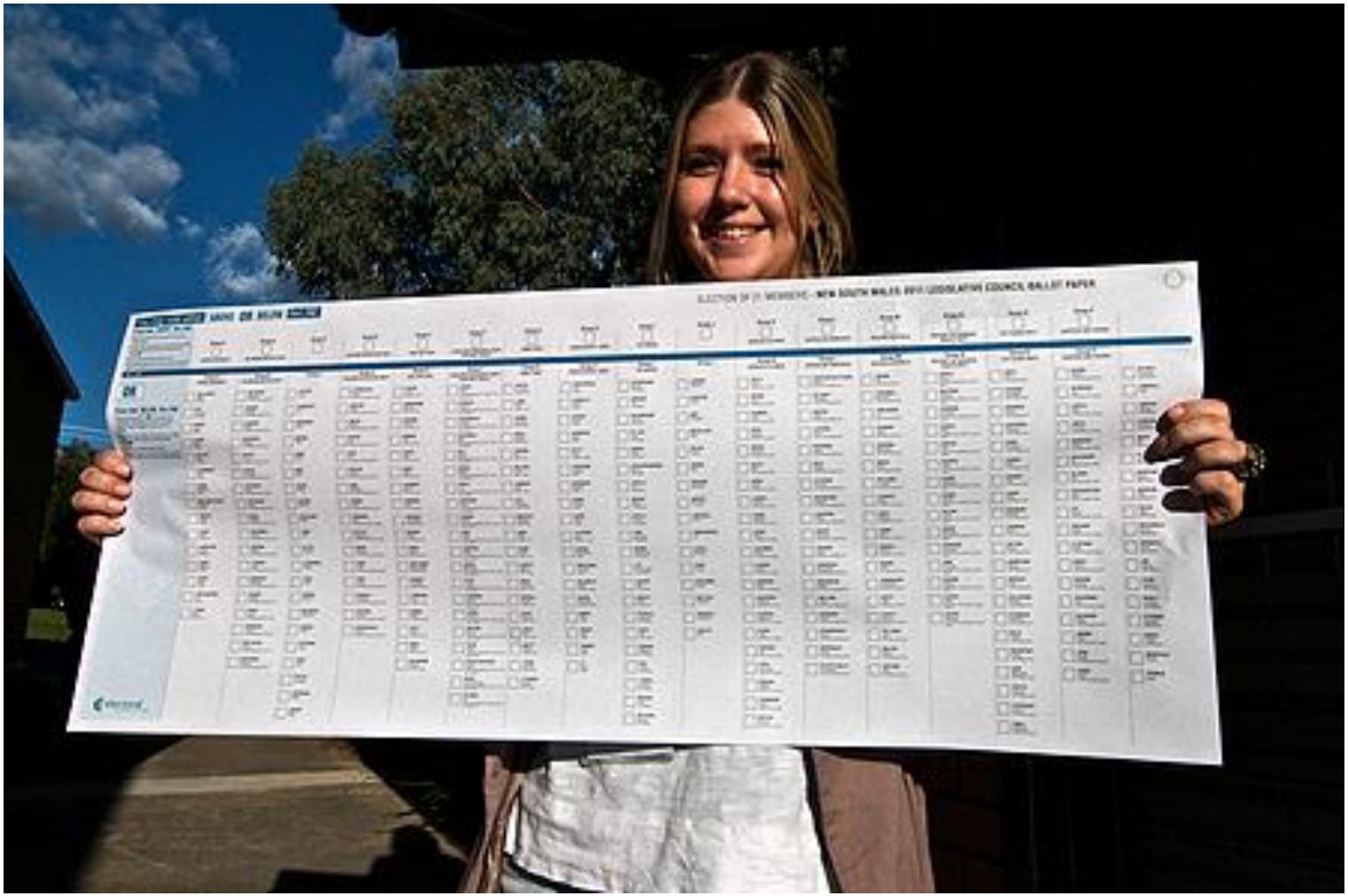
Encrypted Ballot

Voter

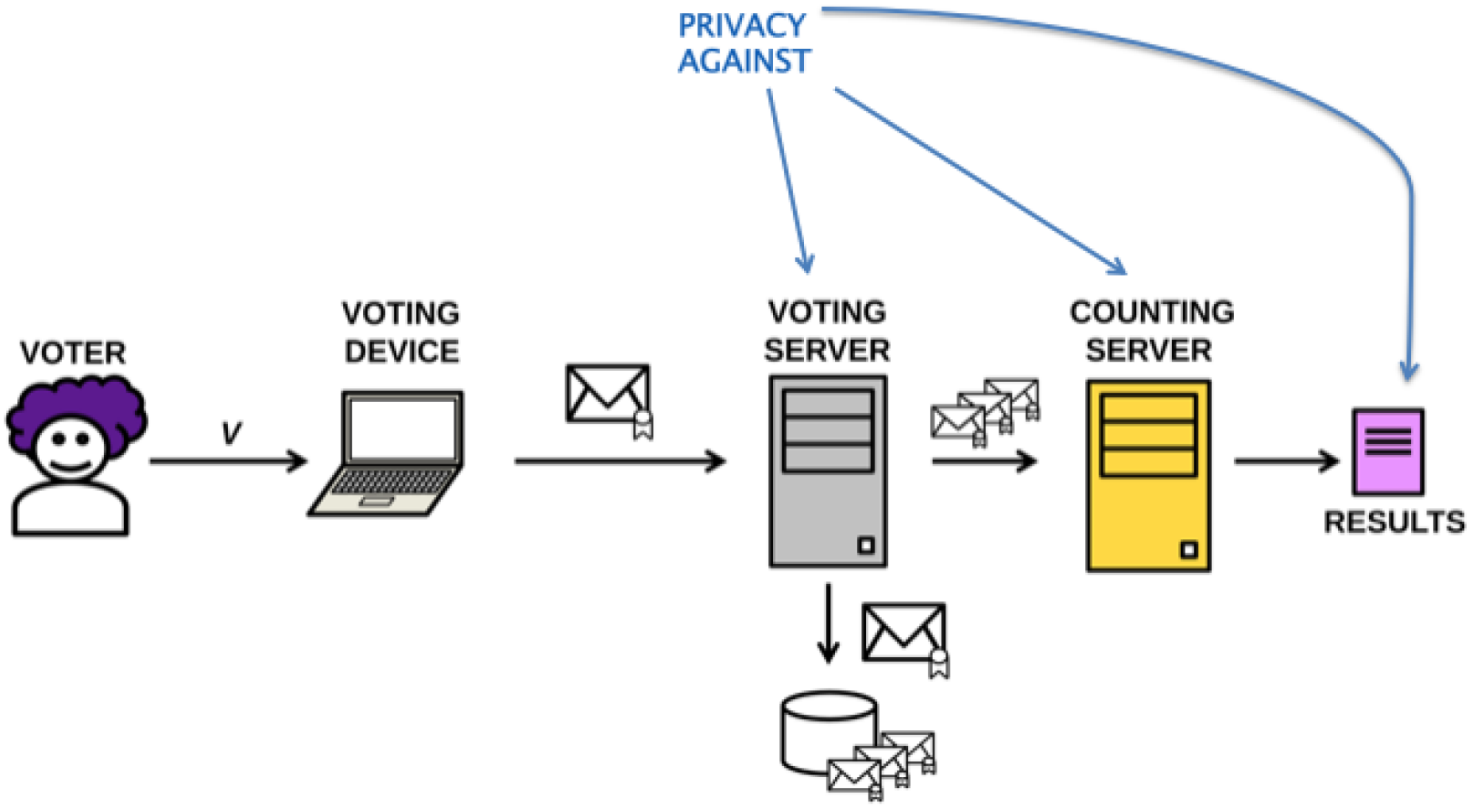
Voting Device

Ballot box

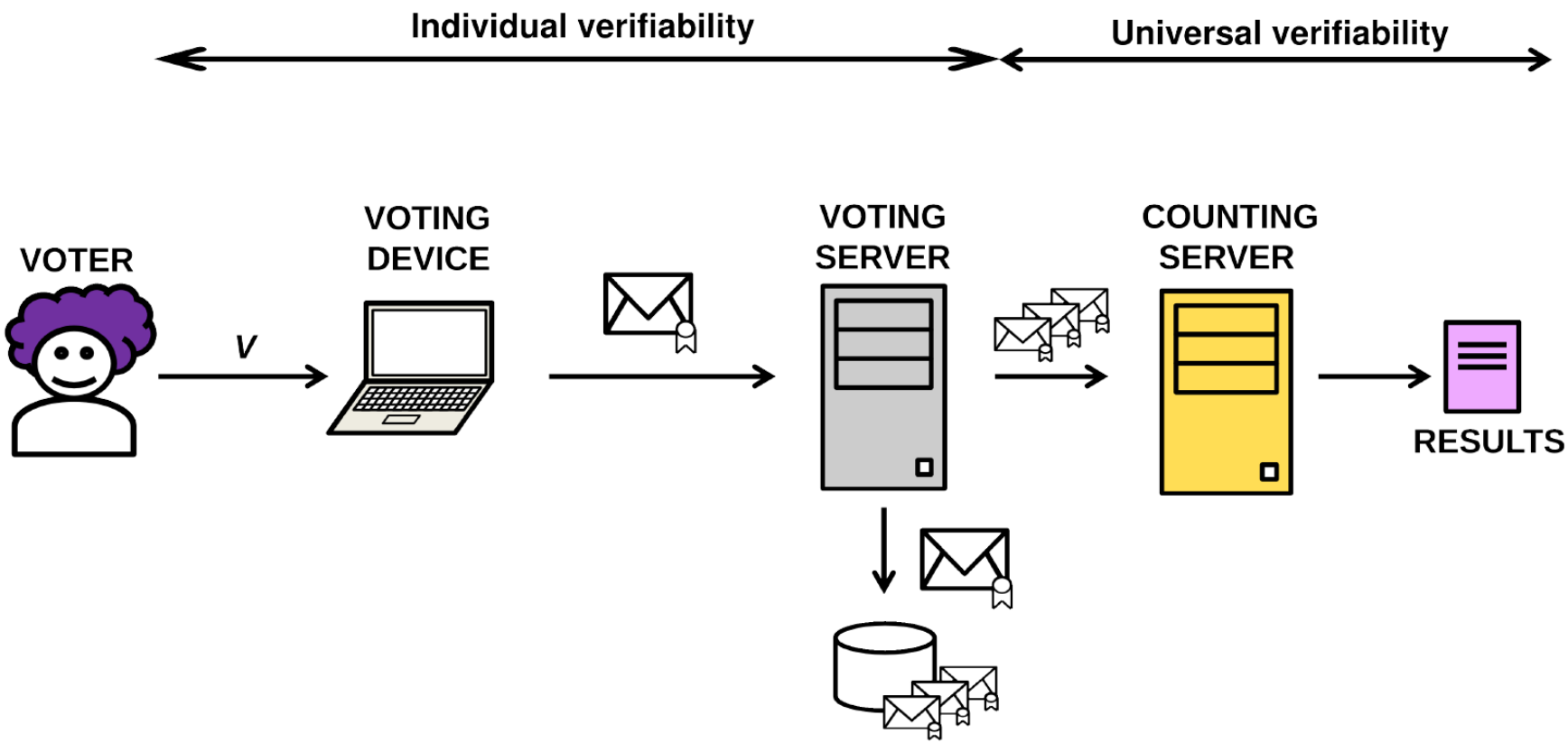
Bulletin Board



Basic Privacy



Types of verifiability



Defining privacy for e-voting

When is a voting scheme private?

How to certify that a crypto e-voting protocol respects **vote privacy**?

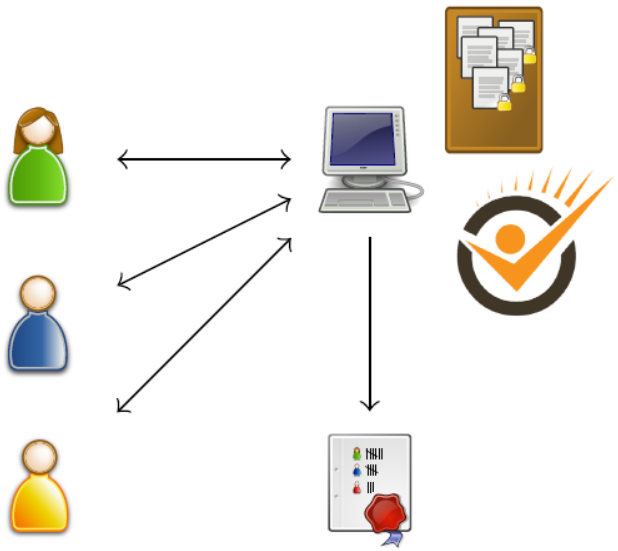
"Doesn't reveal how anyone voted":

too strong!

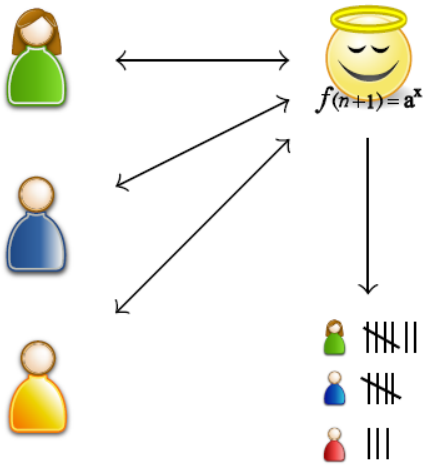
Result: (`yes` 3, `no` 0). How did Alice vote?

Real/ideal world principle

Real world



Ideal world



However, not a simple proof technique…

Entropy

Idea: $H(\text{checklist} \mid \text{locked docs}, \text{sealed checklist}) = H(\text{checklist} \mid \text{people})$

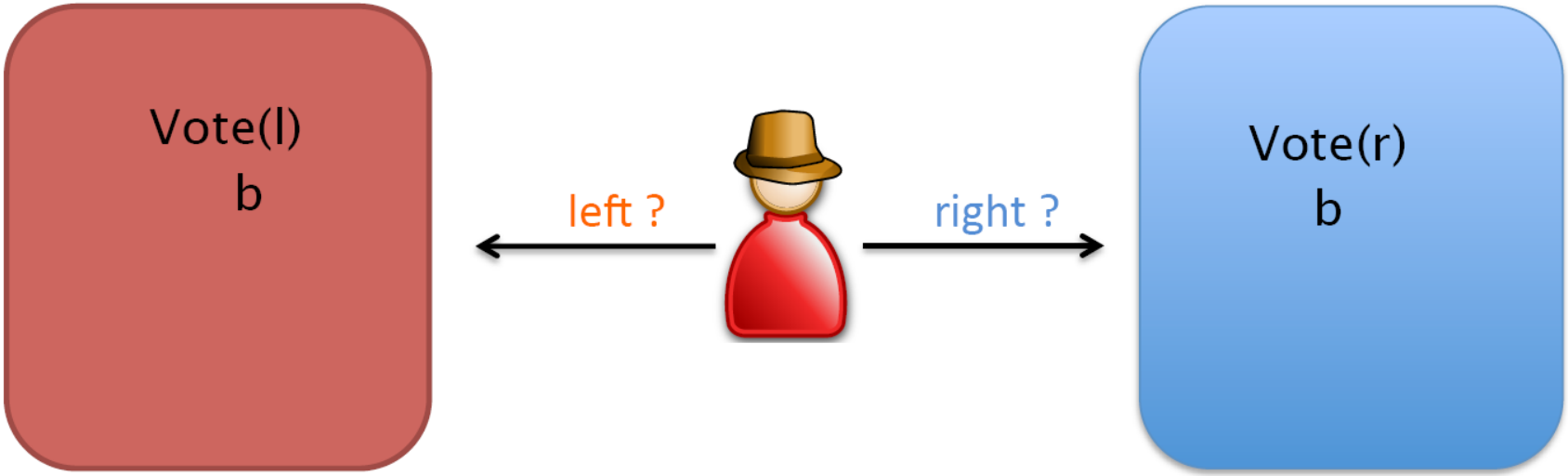
Problem: $H(\text{checklist} \mid \text{locked docs}, \text{sealed checklist}) = 0$

Solution: $H_c(\text{checklist} \mid \text{locked docs}, \text{sealed checklist}) = H(\text{checklist} \mid \text{people})$

Useful metric, but even less simpler proof technique...

Game-based security

Distinguish left from right, publish ????, restrict ????.

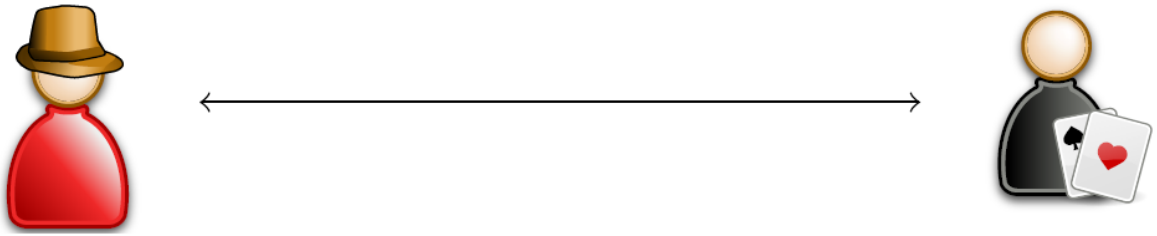


1. Obtain pk.
2. In any order:
   - Submit two votes (l, r).
   - Submit a ballot b.
   - Read board.
3. Ask for tally.

1. Provide pk.
2. Answer with:
   - ballot for l/r vote.
   - check ballot.
   - board.
3. Return ????

Game-based security ESORICS 2011

Distinguish left from right, publish left tally, no restriction.

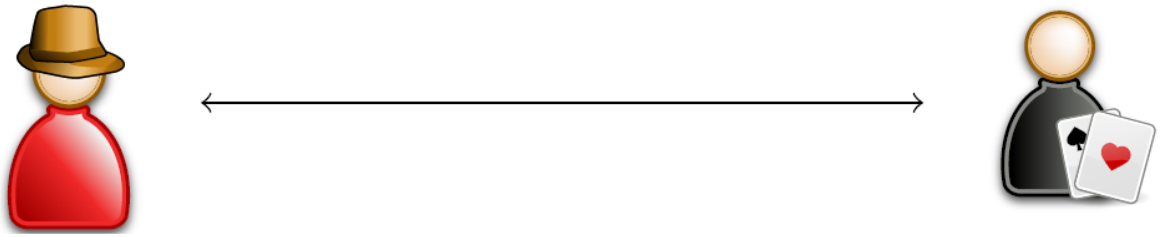


1. Obtain pk.
2. In any order:
   - Submit two votes (l, r).
   - Submit a ballot b.
   - Read board.
3. Ask for tally.

1. Provide pk.
2. Answer with:
   - ballot for l/r vote.
   - check ballot.
   - board.
3. Return *left* tally.

Game-based security ESORICS 2011-13, PKC 2015

Distinguish left from right, publish left tally, no restriction.



	LEFT game:	RIGHT game:	
vote (yes , no)	$\{\text{yes}\}_{pk}$	$\{\text{no}\}_{pk}$	yes
vote (yes, no)	$\{\text{yes}\}_{pk}$	$\{\text{no}\}_{pk}$	yes
vote (no , no)	$\{\text{no}\}_{pk}$	$\{\text{no}\}_{pk}$	no
ballot ($b = \{\text{yes}\}_{pk}$)	b	b	
tally	(3 yes, 1 no)	(3 yes, 1 no)	

Privacy and verifiability incompatible!

Game-based security PODC 86, STOC 94, JCS 09

Distinguish left from right, publish real tally, on permutation-equivalent honest assignments.


yes


no

≈


no


yes

vote (yes, no)
vote (no, yes)
tally

LEFT game:

$\{yes\}_{pk}$
$\{no\}_{pk}$

(1 yes, 1 no)

RIGHT game:

$\{no\}_{pk}$ yes
$\{yes\}_{pk}$ no

(1 yes, 1 no)

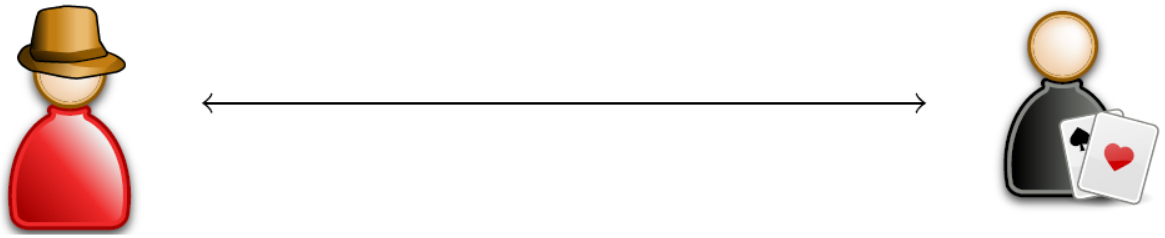
Game-based security PODC 86, STOC 94, JCS 09

Distinguish left from right, publish real tally, on permutation-equivalent honest assignments.

Swiss, Luxembourg elections not covered!

Game-based security IEEE S&P 2015

Distinguish left from right, publish **left result**, publish **simulated tallying proofs**, no restriction.

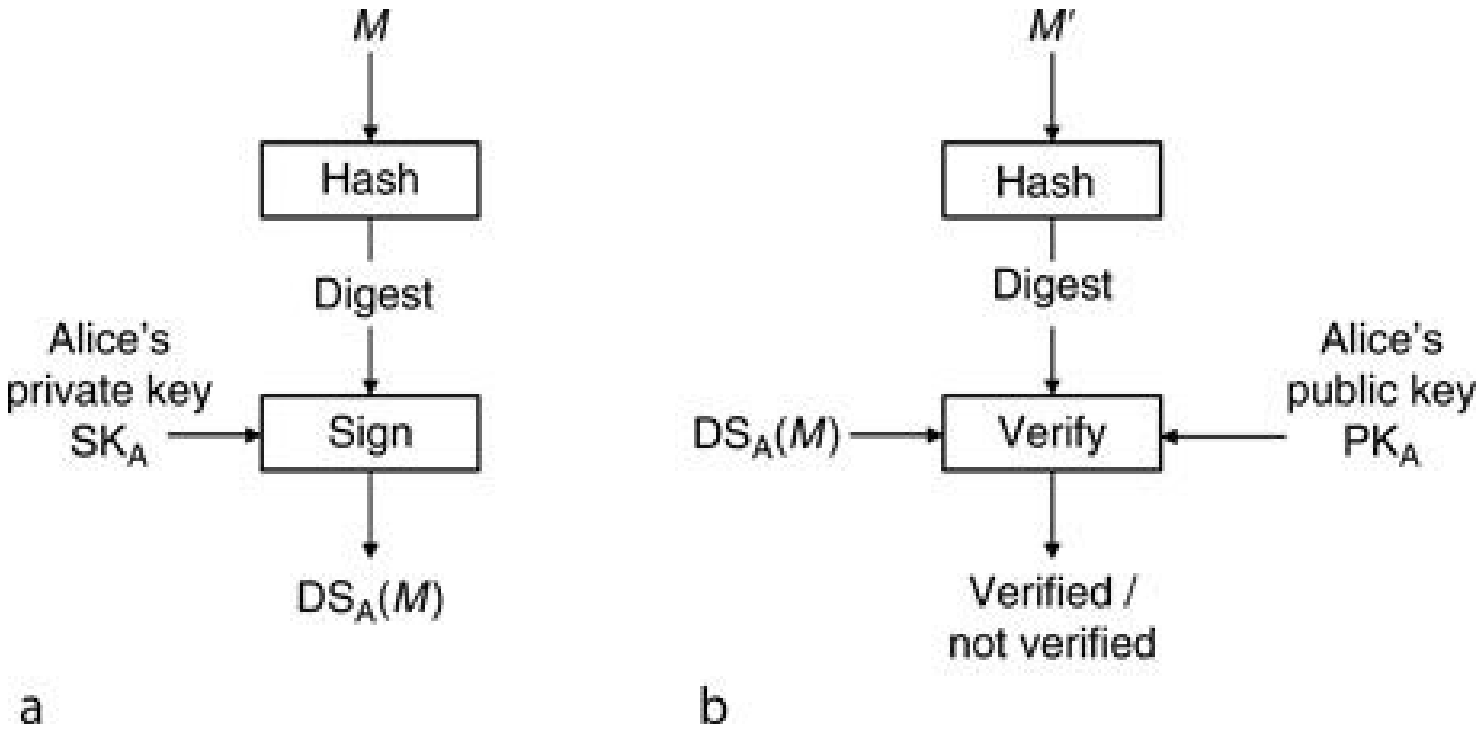


1. Obtain *pk*.
2. In any order:
   - Submit two votes (l, r).
   - Submit a ballot *b*.
   - Read board.
3. Ask for tally.

1. Provide *pk*.
2. Answer with:
   - ballot for *l*/*r* vote.
   - check ballot.
   - board.
3. Return *left* tally, *simulated* proofs

E-voting:
canonical designs

Digital Signatures



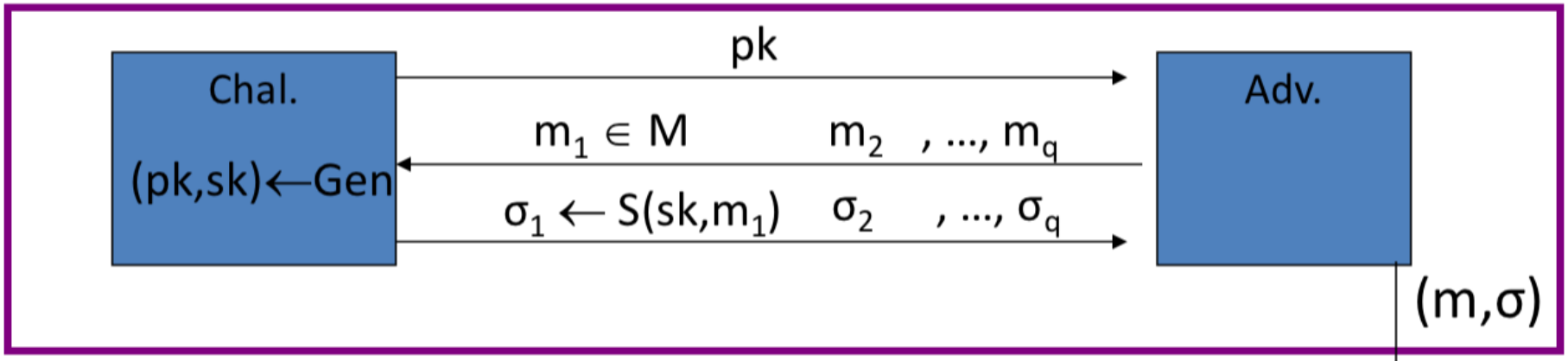
Digital Signatures

A digital signature $\mathcal{S}$ consists of three algorithms $(\mathrm{KG}, \mathrm{Sign}, \mathrm{Verify})$:

- $\mathrm{KG}(\mathrm{pms}(\lambda))$ on input **global parameters** pms outputs a pair of verification/signing keys (vk, sk), whereby vk is public, and sk is secret to the signer
- $\mathrm{Sign}(sk, m; r)$ on inputs a verification key vk, string m and (possibly) randomness r outputs a signature σ
- $\mathrm{Verify}(vk, m, \sigma)$ on inputs a verification key vk, a string m and a signature σ, outputs yes/no, whereby yes means that σ is a valid signature on the digital document m

Unforgeability

For a sig. scheme (Gen,S,V) and adv. A define a game as:



Adv. wins if **$V(pk,m,\sigma)$ = `accept'** and **$m \notin \{m_1, \ldots, m_q\}$**

Def: SS=(Gen,S,V) is **secure** if for all "efficient" A:

$Adv_{SIG}[A,SS]$ = Pr[A wins] is "negligible"

1^{st} attempt: total transparency

Voting Phase

Let $\sigma_X := \text{Sign}(sk_X, v_x)$, where v_x is the voting option of player P_x and (pk_x, sk_x) is the signing key pair of P_x

Ballot Box

Alice	(v_A, σ_A)
Bob	(v_B, σ_B)
Chris	(v_C, σ_C)
Daniel	(v_D, σ_D)
...	...

Tally Phase: $\sum_{x \in X} v_x$ for all v_x such that $\text{Verify}(pk_x, v_x, \sigma_x) =$accept

… but no privacy!

- **Privacy**
  - To know how a voter X voted, an adversary locates X's ballot $b_X := (v_x, \sigma_x)$ and learns voting choice v_X
  - Hence this system does not achieve **privacy**

2nd attempt: adding encryption

Voting Phase

Let E be a non-malleable PKE scheme. Let (pk_E, dk_E) the encryption key pair for the election. Let $c_x = \text{Enc}(pk_E, v_x)$ be the encryption of the voting choice v_x. Let $\sigma_x = \text{Sign}(sk_x, c_x)$

Ballot Box

Alice	$(\text{Enc}(pk_E, v_A), \sigma_A)$
Bob	$(\text{Enc}(pk_E, v_B), \sigma_B)$
Chris	$(\text{Enc}(pk_E, v_C), \sigma_C)$
Daniel	$(\text{Enc}(pk_E, v_D), \sigma_D)$
...	...

Tally Phase $\sum_{x \in X} v_x$ for all $v_x := \text{Dec}(dk_E, c_x)$ such that $\text{Verify}(pk_x, c_x, \sigma_x) = \text{accept}$

Privacy

- Assumptions:
  - Voting device VD is trusted
- From X's ballot $b_X := (\mathrm{Enc}(pk_E, v_x), \sigma_x)$ an adversary cannot learn voting choice v_x if the PKE scheme is non-malleable
- Tally only publishes the end result $\sum_{x \in X} v_x$
- An adversary can only learn partial information about v_X from the election result!

but no verifiability!

Canonical design for a ‘yes’/‘no’ election

Voting Phase:

Ballot Box		
Alice	$\mathrm{Enc}(pk_E, v_A)$	$\mathrm{ZPK}\{v_A = 0 \text{ or } 1\}$
Bob	$\mathrm{Enc}(pk_E, v_B)$	$\mathrm{ZPK}\{v_B = 0 \text{ or } 1\}$
Chris	$\mathrm{Enc}(pk_E, v_C)$	$\mathrm{ZPK}\{v_C = 0 \text{ or } 1\}$
Daniel	$\mathrm{Enc}(pk_E, v_D)$	$\mathrm{ZPK}\{v_D = 0 \text{ or } 1\}$
...	...	

Phase 2: Tally - homomorphic encryption (ElGamal)

$$\prod_{i=1}^{n} \mathrm{Enc}(pk_E, v_i) = \mathrm{Enc}(pk_E, \sum_{i=1}^{n} v_i)$$

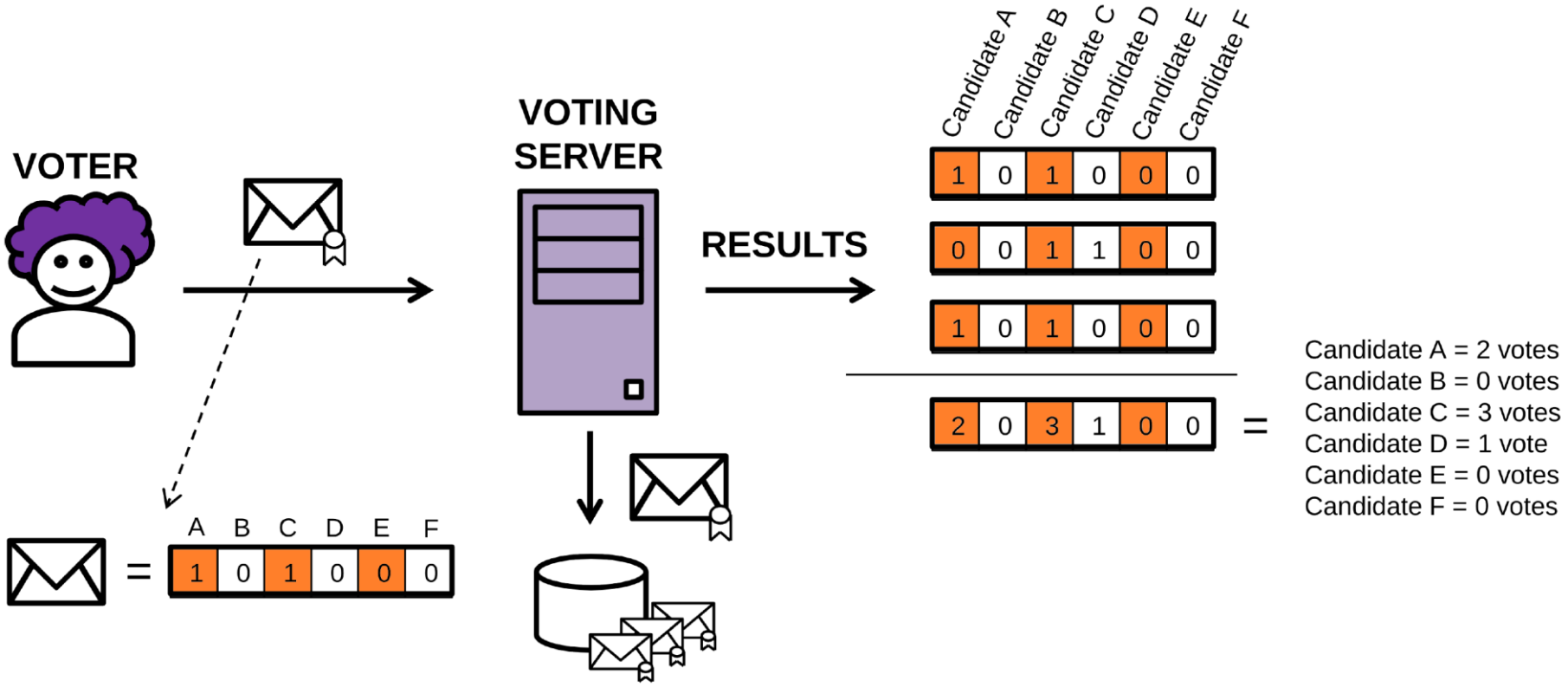
relies on $g^a \times g^b = g^{a+b}$

$\rightarrow$ Only the final result needs to be decrypted!

$\rightarrow$ Correct decryption proven by using ZKP.EqDL !

pk_E: the corresponding decryption key dk_E is held by the tallying authority

Homomorphic tallying



Models and definitions not so neutral…

values, tastes, judgments, …

Disciplinary culture

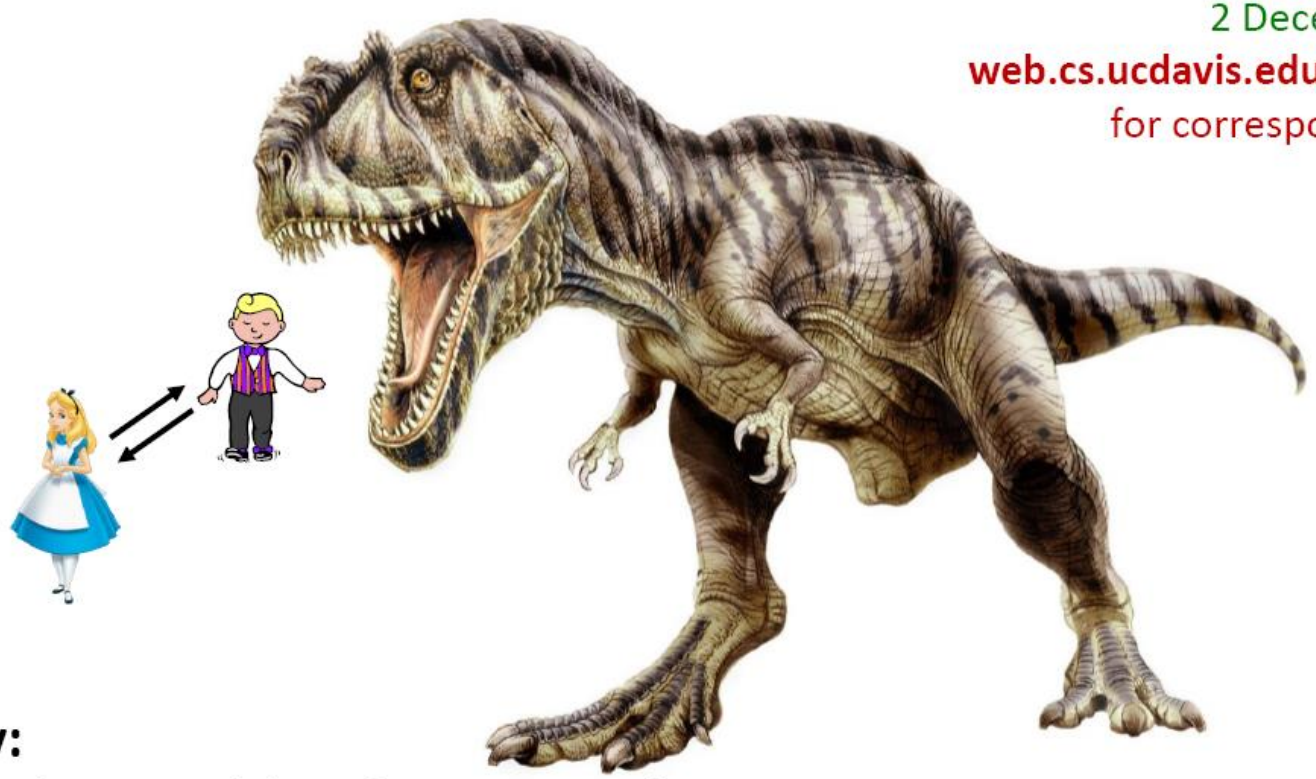
Papers

The Moral Character of Cryptographic Work

Phillip Rogaway

IACR Distinguished Lecture
Asiacrypt 2015
Auckland, New Zealand
2 December 2015

web.cs.ucdavis.edu/~rogaway/
for corresponding essay



Today:

① Social responsibility of scientists and engineers
② The political character of cryptographic work
③ The dystopian world of pervasive surveillance
④ Creating a more just and useful field

Motivations (conflicting?)

€815,000 spent for 525 voters and their dependents to fly to Malta to vote in MEP elections

Albert Galea Monday, 28 October 2019, 09:01 Last update: about 12 days ago

Backed by Scytl, UAE Continues to Innovate in Hosting Third Fully Electronic Parliamentary Elections

– 117,592 voters cast electronic ballots in 2019 Federal National Council elections
– Voter participation increases 48.5% over 2015
– Results announced in under 15 minutes

Why do disabled people feel ignored when it comes to voting?

By Kathleen Hawkins
BBC News, Ouch

Share

UP FRONT

India's electoral democracy: How EVMs curb electoral fraud

Madhavan Somanathan · Friday, April 5, 2019

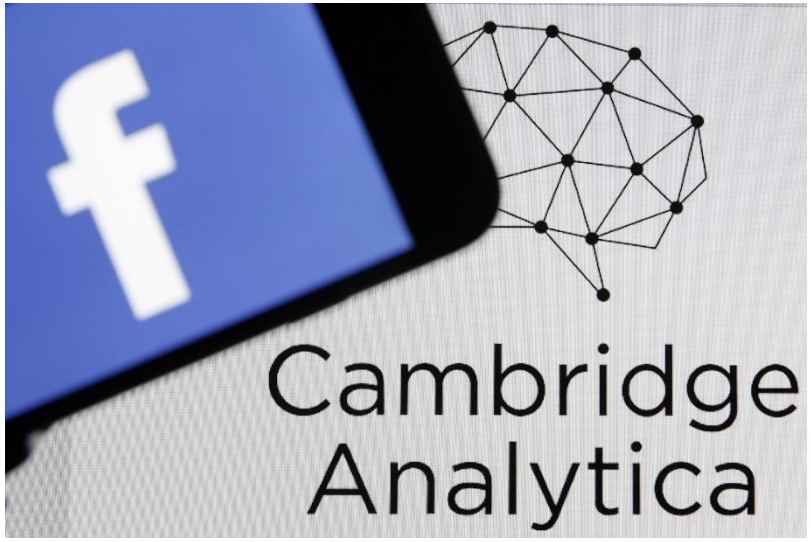
Electronic voting: the controversy

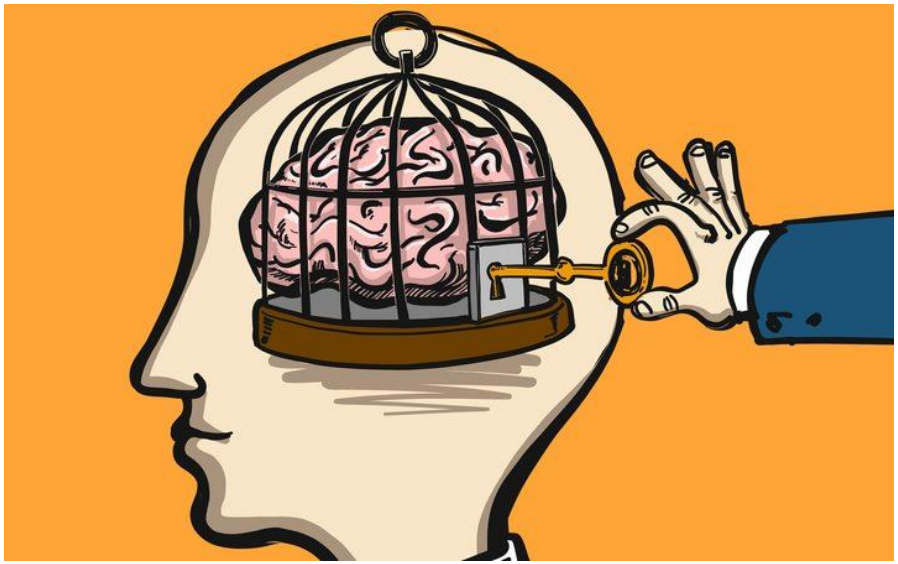
Main argument is that *it will be used* to

rig elections

How to rig an election ***today***

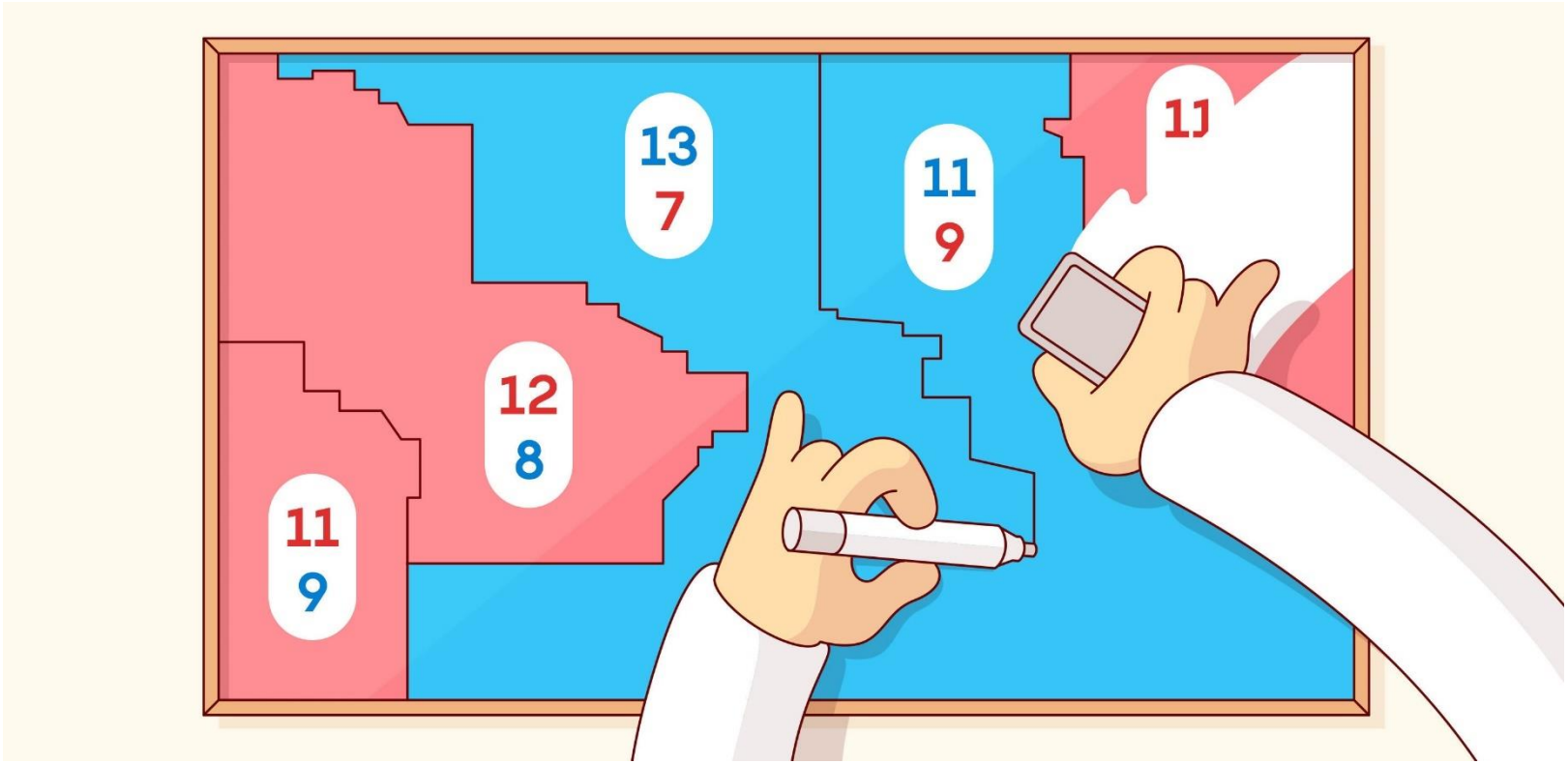
Misinformation and/or Propaganda





How to rig an election ***today***

Gerrymandering



How to rig an election ***today***

Voter suppression

Opinion

Voter ID checks threaten to suppress voter rights 'on an industrial scale'

There were only 8 cases of voter ID fraud in 2018, there are more serious threats to our democracy that need to be addressed

By Darren Hughes
Monday, 14th October 2019, 9:21 pm
Updated Wednesday, 16th October 2019, 11:15 am

A tool for preventing voter suppression

An **election result** can be **radically different** if you manage to **suppress**/**encourage** certain voter groups

Demographics **failed currently** by the voting system:

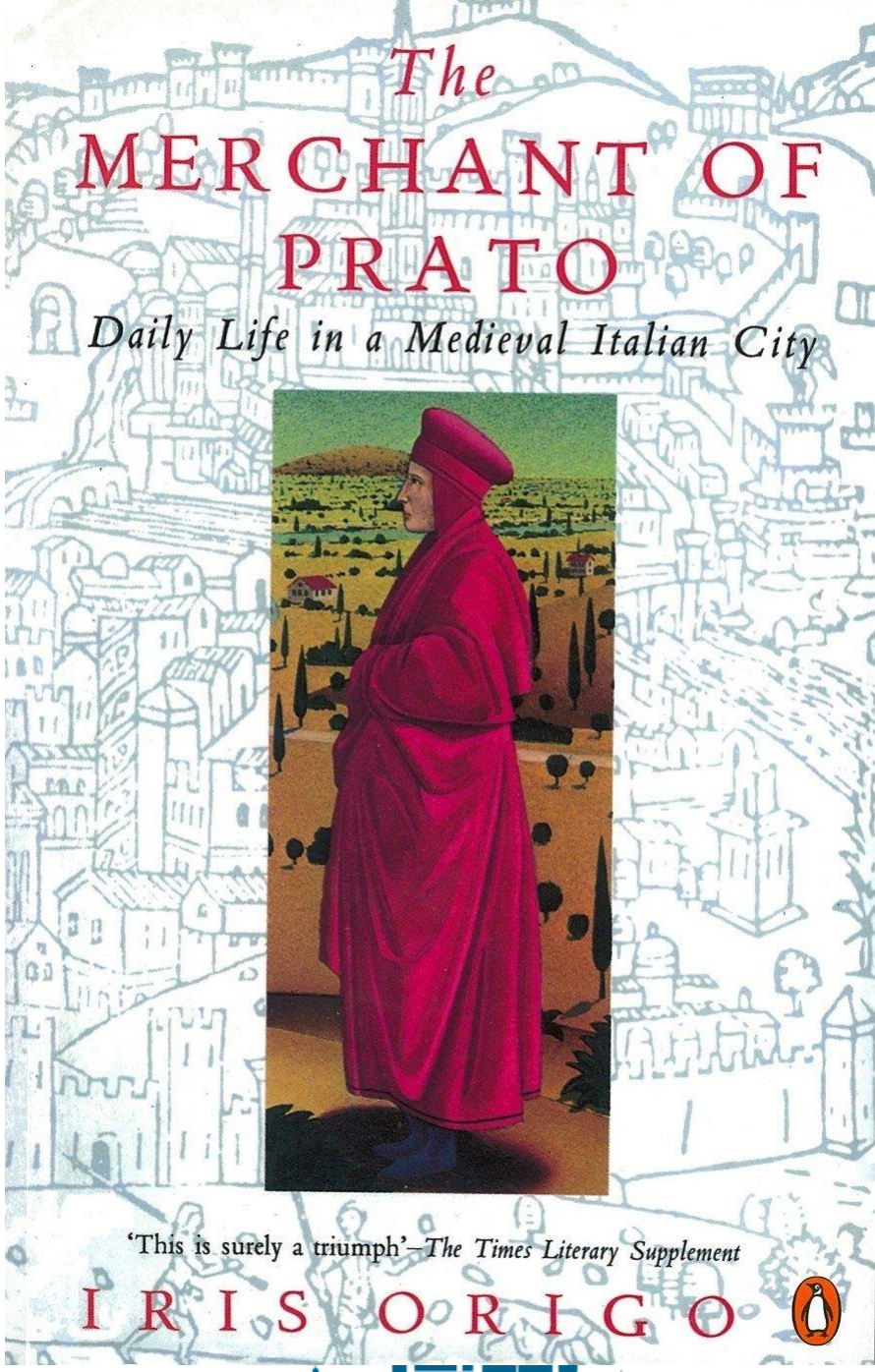
- voters with disabilities:
  - vision impairments, reduced mobility, mental illness
- expatriates voters
- overseas military voters

Recap: we are voting using a **19th century** tools**!**

Blockchain

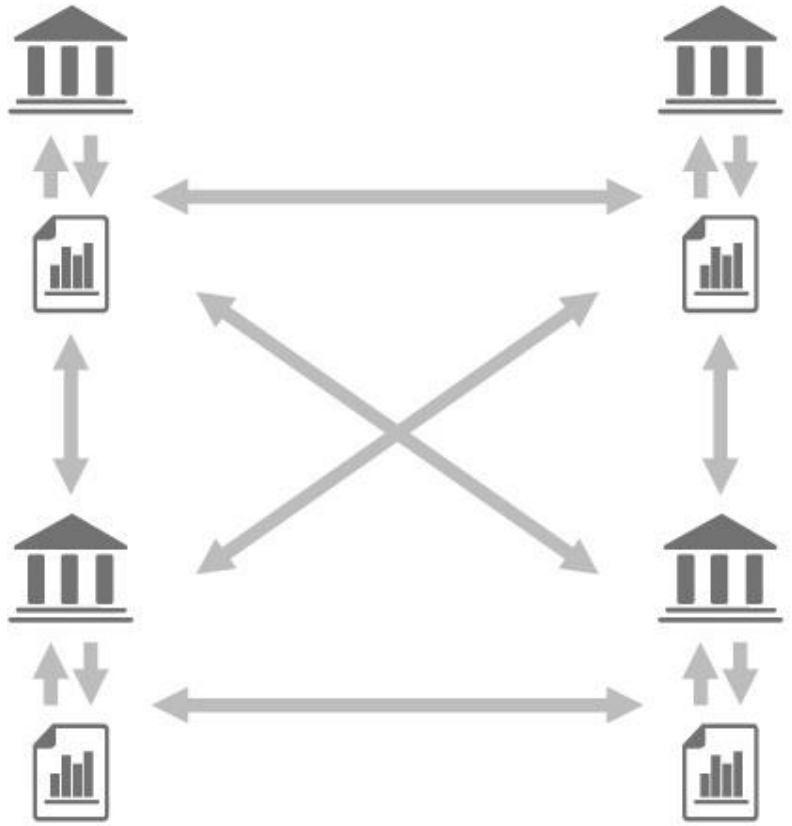
It‘s All about Ledgers

- A **ledger** is a sequential list of transactions
- Examples of ledgers:
  - financial assets (fiat currency!)
  - banking transactions
  - academic certificates
  - land registry

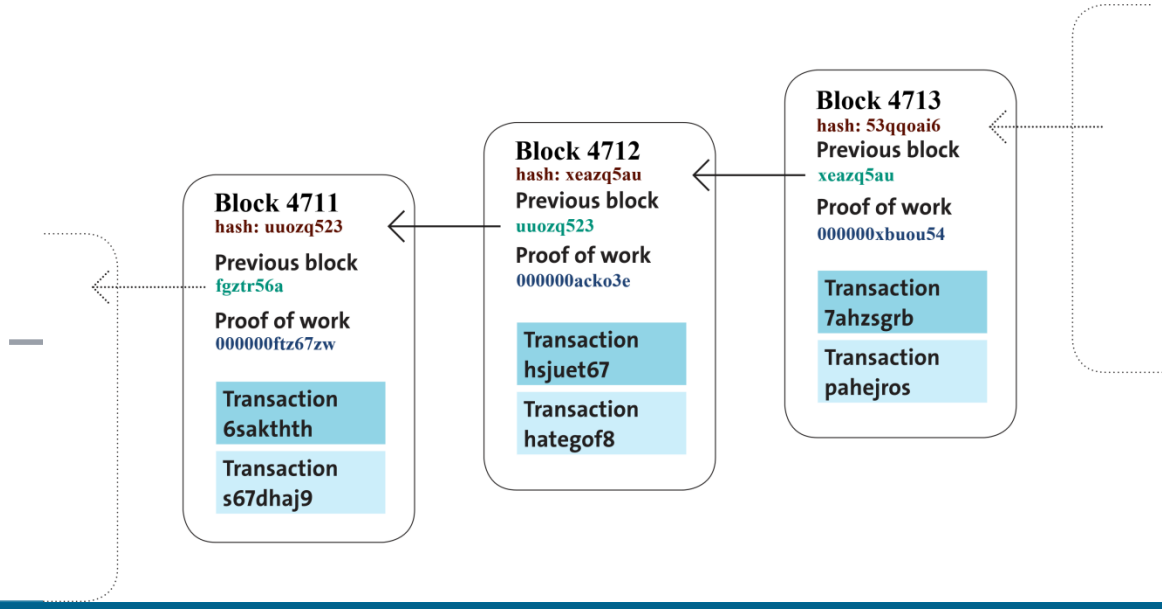


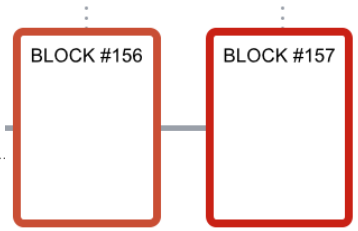
What is a Blockchain (DLT)?

- A **distributed ledger** for digital assets ensuring:
  - immutable and time-stamped entries
  - provenance
  - replication
  - consensus



distributed ledger

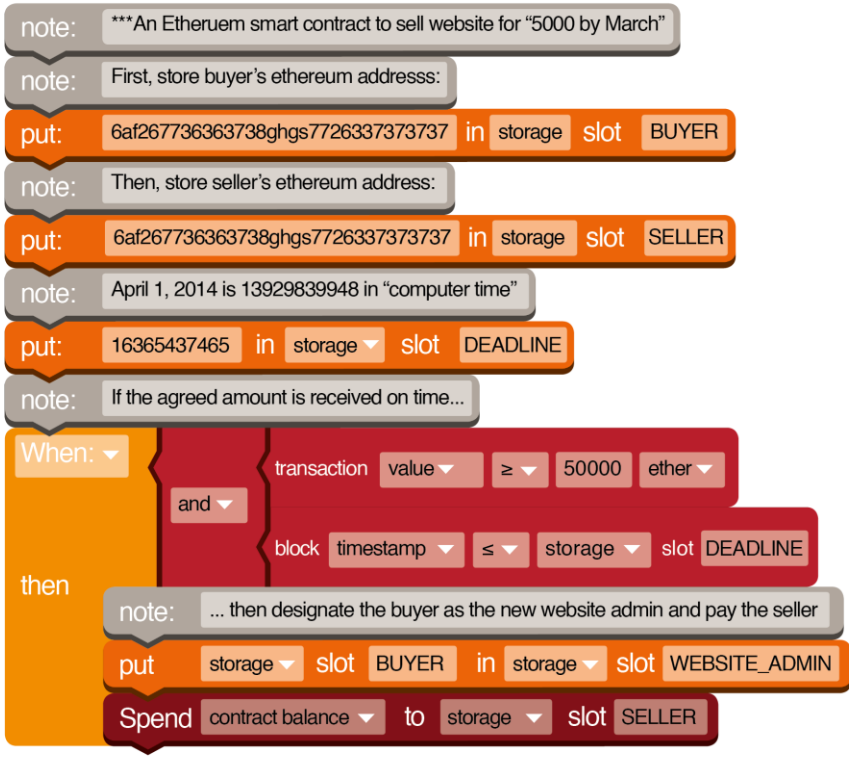




Smart contracts

"a digitally signed, computable agreement between two or more parties"

A **software agent** executes and enforces the terms of such agreements



Blockchain Zoo

ethereum

CASH

HYPERLEDGER

fetch.ai

MONERO

c·rda

IOTA

The Coco Framework

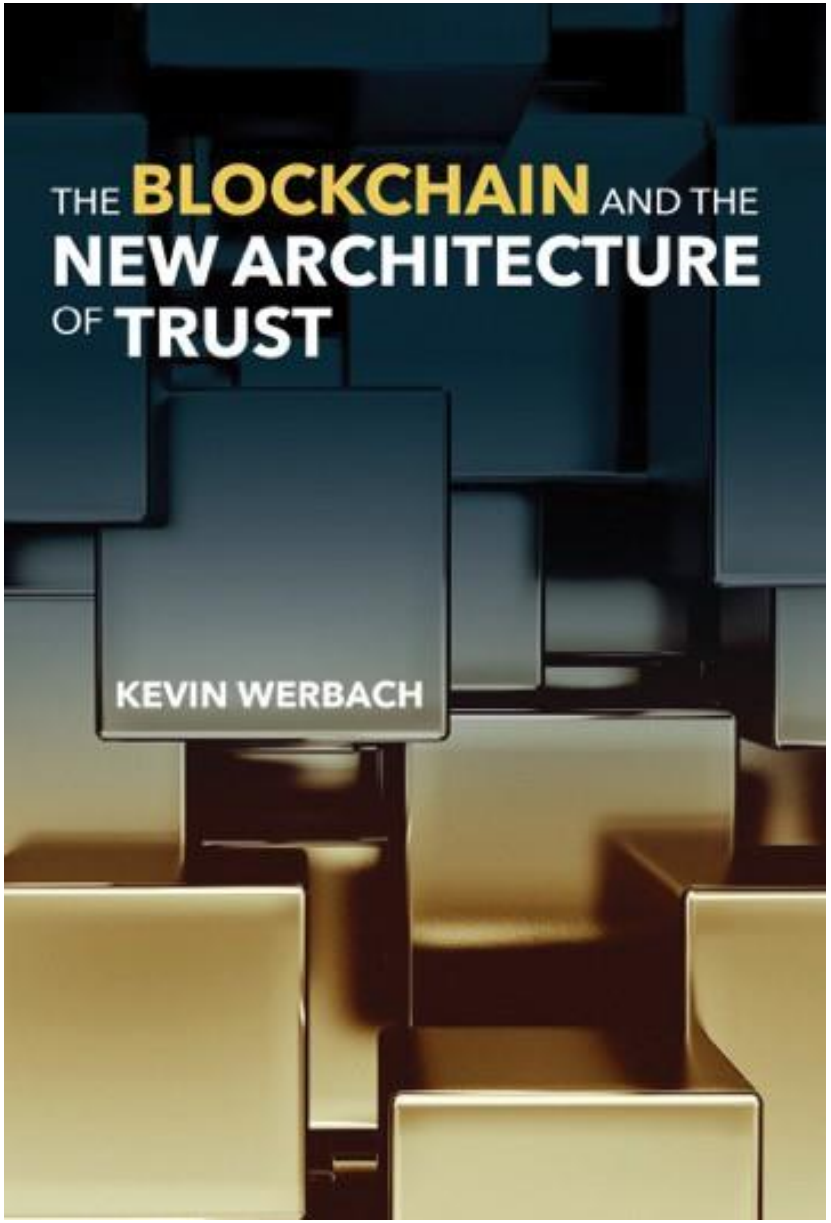
Powered by Microsoft

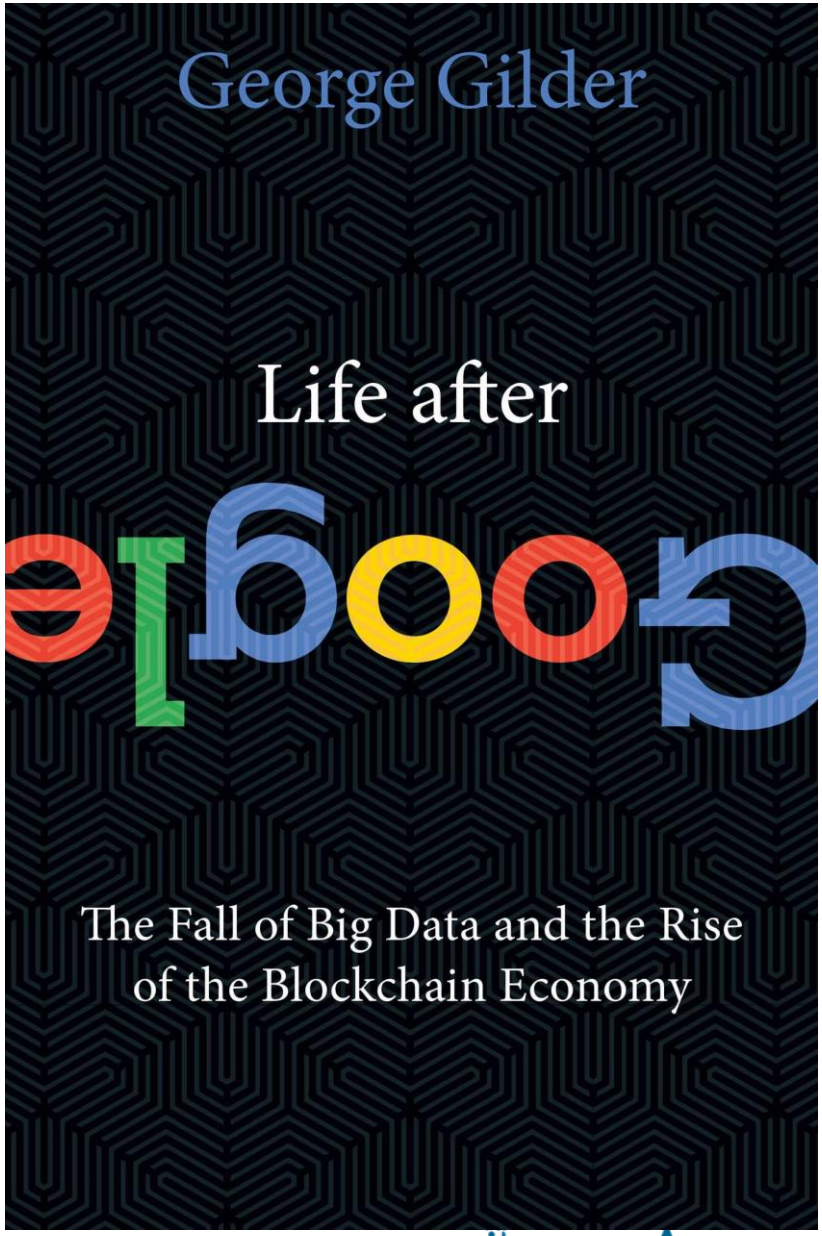
What are DLTs good for?

Distributed Ledgers are technological **tools** that facilitate social/financial **interactions** between **strangers:**

- DLTS are not an end in itself
- Useful in the presence of **distrustful partners** or
- where a ***trust gap*** exists among coordinating entities

DLTs are seen as technological **replacements for mediators** and to **decrease friction** in multi-party systems

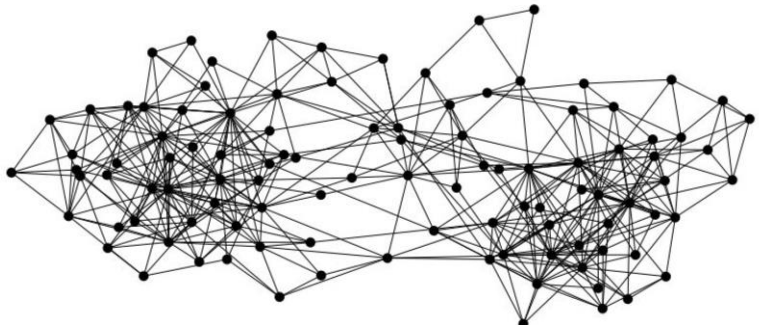

THE BLOCKCHAIN AND THE
NEW ARCHITECTURE
OF TRUST
KEVIN WERBACH


George Gilder
Life after
Google
The Fall of Big Data and the Rise
of the Blockchain Economy

Connections: e-voting and blockchain

signed by Alice

Pay to pk_{Bob} : H( )



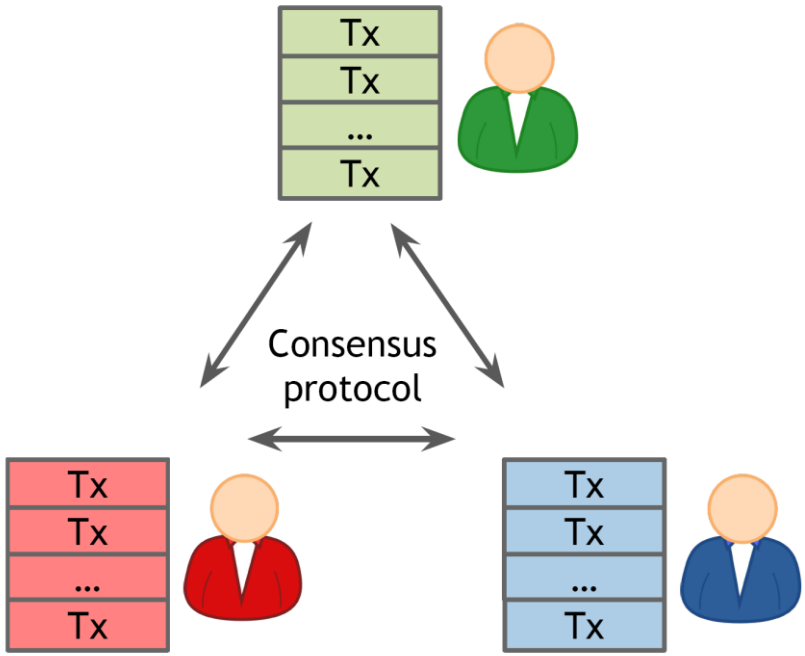
Ballot Box

Alice	(v_A, σ_A)
Bob	(v_B, σ_B)
Chris	(v_C, σ_C)
Daniel	(v_D, σ_D)
...	...

zkSNARKs

zero-knowledge Succinct Non-interactive ARguments of Knowledge

Achieving consensus

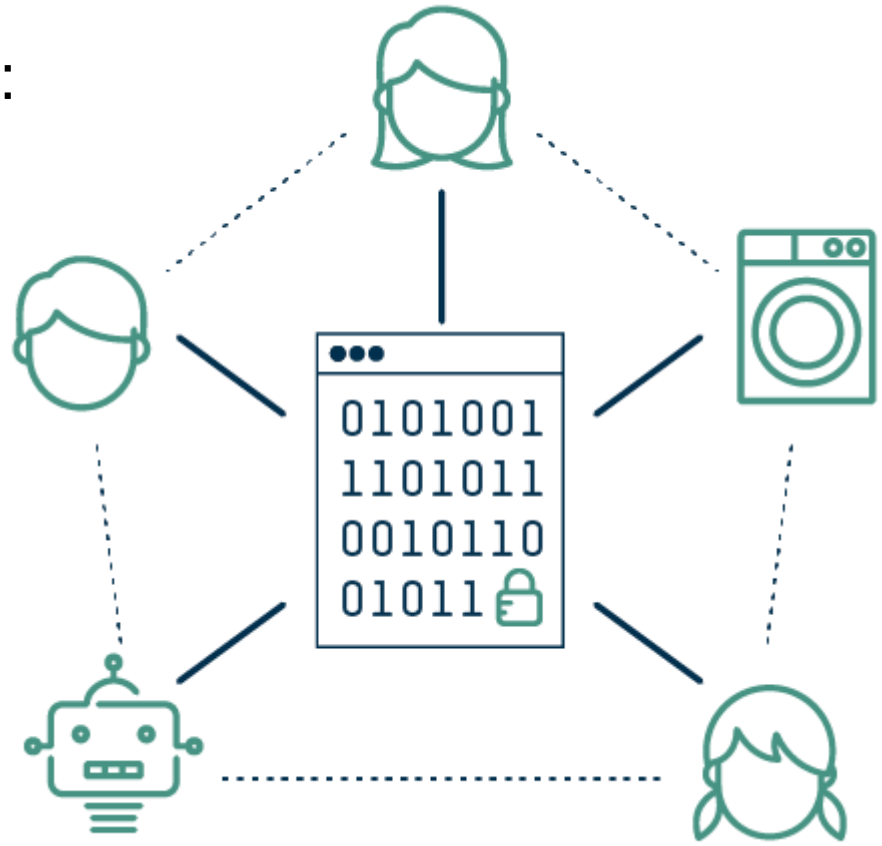


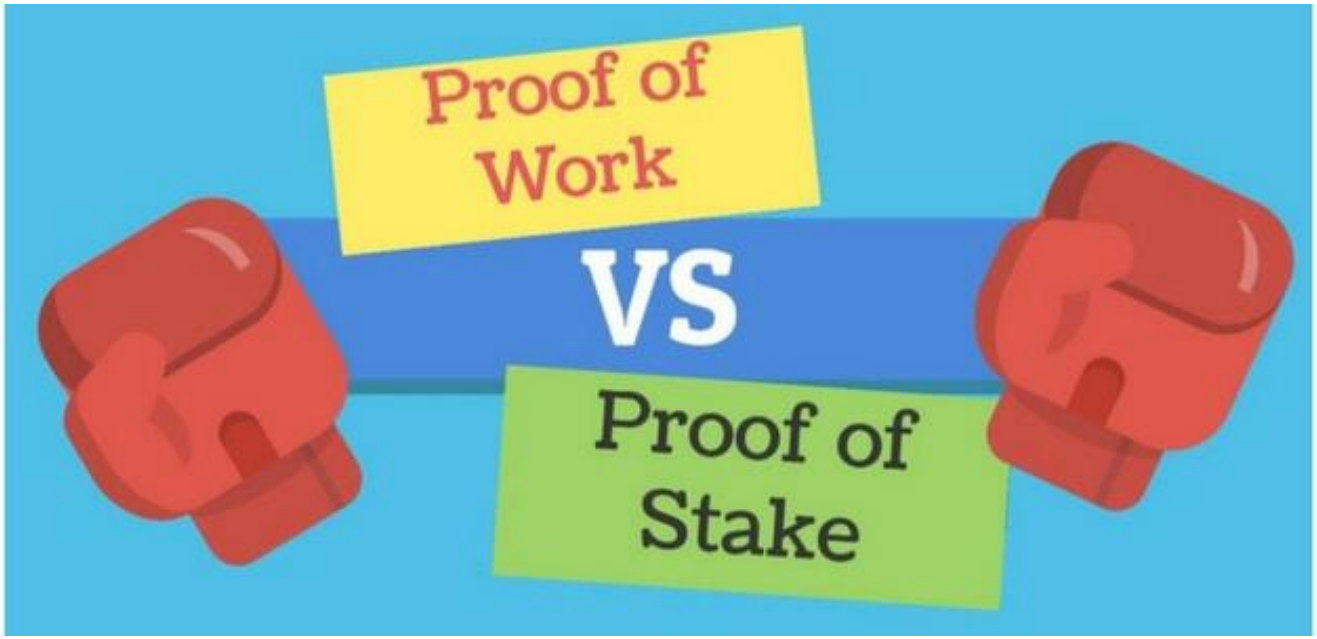
Choosing a leader in a distributed decentralized network

Threshold Cryptography

Distributed random computation:

- Unbiased
- Pseudorandom
- “Unstoppable”
- Low overhead




Proof of Work
VS
Proof of Stake