Lecture Notes
electronic elections

Michael Nüsken

b-it

(Bonn-Aachen International Center
for Information Technology)

winter 2007/2008

Times:

Course ✓

Tutorial:

	Mon	Tue	Wed	Thu
8	4	3		
10	4			
12	8	8		
14	Tutorial 4	5	4	
16	Malware	4	4	
18	1/2	2	3	

Election?

- choose an option
- position in government
- somebody votes
- head/leader of organization (state/company)

} Make a decision!

One solution: a dictatorship.

elect²
31.10.07
2

Democracy:

- determine leader(s) in a fair way.
- people decide about the leader(s)
- people decide about "state affairs", important questions.

Necessary properties of an election in these circumstances

- free and fair.

Germany:

frei – free

gleich – equal, "one-man-one-vote"

geheim – secret

allgemein – universal (no restriction by race, gender, belief, social status)

direkt – direct

elect²
31.10.07
3

Election: Means to determine
the political will of the people
or: to make decisions.

Non-issue: <u>Not</u> to form political opinions.
Though...

Legal conditions: Laws define elections.

→ when elections take place,
which questions are decided.

→ which principles hold for the election,
in particular: who can vote

→ how the votes are evaluated
and combined into an answer

Election:
- Registration
- Election itself
- Counting

elect[2] 7.11.07 (1)

Survey of ~~techno~~ voting technology

- Australian paper ballot

≈ -750 ... -146	Ancient Greeks	pieces of broken pottery
-139	Romans	'gabinia lex' → (paper) ballots for elections of magistrates
1629	MA Bay Colony	Election of a pastor for the Salem Church
1795	France	French constitution states in Art. 31 'Toutes les élections se font au scrutin secret.' (All elections are to be held by secret ballot.)

Ballot — 1549, from it. pallotte, diminuitive of palla "ball". Earliest reference due to Venice.

1838	Britain	Chartist petition among other things asks for secret ballots.
1854	Australia	Influenced by Chartists, revolting miners in Victoria adopt secret ballot (as part of the entire Charta).
1856	Australia	States Tasmania, Victoria and South Australia enact legislation. New South Wales (1858), Queensland (1859)

Western Australia (1877) followed.

elect² 7.11.07 ②

1870 New Zealand

1872 Britain 'Ballot Act' introduces secret ballots.
This reduces substantially the cost of campaigning.

1874 Canada

1884-91 USA All states move to "Australian ballot" (from oral ballot)

- an official ballot is printed at public expense,
- on which the names of the nominated candidates of all parties and all proposals appear,
- being distributed only at the polling place and
- being marked in secret.

The first US presidential election under secret ballot took place in 1892.

1901 Denmark

1920? Germany

Pro / Cons

elect^c
7.11.07
3

Pro: Secrecy implies 'free' votes.
· A voter cannot sell his vote,
because he has no proof of his vote.

· 'Family' voting also becomes impossible.

· No corruption is effective,
as long as the principles hold.

(Short viewed pros it's cheaper, no campaigning...)

Necessity: Ballots must always be handled by an official under supervision by someone representing an opposing party. Because of this, the partisan affiliations of each election official must be declared in advance.

Con: Counting? When is a mark valid?
If the counting team is biased they may declare unwanted votes as invalid.

Con: Bad layout? See 2000 US presidential election.

7.11.07
④

Lever voting machines

1892: NY, first use

...1930: all larger urban centers

Pro / Cons

Pro: No bias in counting,
no invalid votes.

Pro: Instant election result.

Pro/Con: Complete voting place violation
(manipulating counters at read-off, ...)

Con: No backups.
Recounting impossible.

Con: Mechanical failures may go unnoticed.
Very complex, testing rarely complete,
only technicians could check.

Punch card voting

1964: IBM's Portapunch punch mechanism,
first in Georgia.

1972: $\approx 10^6$

1998: roughly 1/3.

Pro: ~~No~~ bias in counting. (?)
Pro: Fast election result.
Pro: Backup available.
Con: Punch often not clean. (1.5%)
Con: Prepunched chads or badly punched ones
may fall, especially during recounting.
Con: No intuitive way to decide whether a vote is valid

Con: Punchcards are often too small, in particular they do not contain names but only 3-digit numbers or so.

Optical Mark-Sense Voting

— ≠

1970s ...
1998: about 1/4 of US voters used this.

⋮

Pro: One can very large voting sheets (up to ~A3)
→ so intuitive interpretation possible

Accuracy: Manufacturers: 10^{-7}. (for perfectly marked sheets)

2000 Florida election: 1 in 2000 votes were problematic
.05%.

⇒ Human factor much more important

Con: Computer based...

Direct Recording Electronic Voting (DRE)

Similar to Lever voting machines.

- Pro: No bias, clean votes, no invalid votes.
- Con: Backup problematic.

Counting

Precinct (ger: Bezirk)

elect^2
14.11.07
1

	Precinct count	Central count
type	decentralized	centralized
supervision	difficult	easy
counting observer	necessarily allowed at each precinct	easy
transmission	accuracy?	security? [ballot box theft, ...] fast?
(overvotes	not a problem	hand examination)

Voting paradox

Arrow's theorem / ...

There is no 'public welfare function'

$$\{ \text{~~vector~~ set of votes} \} \longrightarrow \text{Ranking}$$

such that the following hold:

- no dictator
- ~~some a~~ each alternative is relevant
- ~~Ranking is shift valid~~
  no tactical voting

elect
19.11.07
2

Remote voting

→ matter of law

→ many enabling technologies used so far.

vote anywhere

On election day you can vote wherever you like.
(Mostly elliminated because of high susceptibility to fraud.)

early voting

E.g. Postal voting
(Counting has to wait until end of election!)

Pro: more possibilities for voters,
more comfortable.
(Hope: higher turnout.)

Con: often many different ballot styles
[Johnson County, Iowa, ~ 100 000 inhabitants (voters)
→ 70 distinct ballot style!]

Con: Fraud is easier
→ Family voting
→ Selling blank absentee ballots
→ ...

Best ~~known~~ defense: Allow revote.

Evaluation

Many different criteria!

- Legal criteria
- Voter acceptance? Usability?
- Whom do we have to trust?
  - How is the system constructed?
  - ... administrated?
  - Open source?
  - Oversight and audit trails?
  - Monitoring (by opposing parties)?
  - Cryptography? Electronic signatures?
  - Modularization? Redundancy?
- Compatibility?
  - Integration into larger systems?
  - Open standards (for ballot formats, ...)
    vs. complete replacement

Many more details once you start to work on details.

elect[?]
14.11.07
4

Cryptographic primitives

Tool: modular arithmetic,
ring of $\boxed{\text{integer modulo } N}$ $(N \in \mathbb{N}_{\geq 2})$

$\mathbb{Z}_N = (\mathbb{Z}_N, +, \cdot)$

$(\mathbb{Z}, +, \cdot)$ $0, 1, -$ z^{-1}

PANIC+
PAN C·
D ENT
...

comm. Ring.

$\{0 \neq 1$

$\mathbb{Z}$ has division with remainder.

Given $a, b \in \mathbb{Z}$ with $b \neq 0$
find $q, r \in \mathbb{Z}$ such that

$$a = q \cdot b + r,$$

r 'smaller' than b: $0 \leq r < |b|$.

Similarly, also univariate polynomials over a field(!).

Inversion in $\mathbb{Z}_N$?
A multiplicative inverse of $x \in \mathbb{Z}_N$ is an element $y \in \mathbb{Z}_N$ such that $xy = 1$.

Example: $x = 5 \in \mathbb{Z}_{11}$. Answer: $y = 9$.

Can we translate the task:

$\exists y : x \cdot_{\mathbb{Z}_N} y = 1$ in $\mathbb{Z}_N$ (*)

to a question in $\mathbb{Z}$?

$\exists y, q : x \cdot_{\mathbb{Z}} y + (-q) \cdot_{\mathbb{Z}} N = 1$ (**)

Question: Can we solve (decide & find) an equation of the form

$$s \cdot a + t \cdot b = 1$$

where $a, b \in \mathbb{Z}$ are given and we want $s, t \in \mathbb{Z}$.

Yes, use the Extended Euclidean Algorithm!

Guideline: try to find s, t such that $\boxed{s \cdot a + t \cdot b}$ is <u>small</u>! (& positive)

$a = 35, b = 11$

Extended Euclidean Algorithm

i	r_i	q_i	s_i	t_i	comment
	$a = 35$		1	0	$35 = 1 \cdot a + 0 \cdot b$
	$b = 11$	3	0	1	$11 = 0 \cdot a + 1 \cdot b$, $35 = 3 \cdot 11 + 2$
	2	5	1	-3	$2 = 1 \cdot a - 3 \cdot b$, $11 = 5 \cdot 2 + 1$
	1	2	-5	16	$1 = -5 \cdot a + 16 \cdot b$, $2 = 2 \cdot 1 + 0$
	0		11	-35	$0 = 11 a - 35 b$

In particular, $16 \cdot 11 = 1$ in $\mathbb{Z}_{35}$.

elect²
21.11.07
1

Thm The EEA computes
given $a, b \in \mathbb{Z}$ (or $\mathbb{F}_q[X]$)
with at most n bits (or degree at most n)
the greatest common divisor g of a and b
and a representation $g = sa + tb$
where $g, s, t \in \mathbb{Z}$ (or $\mathbb{F}_q[X]$)
using at most $O(n^2)$ bit operations
(or $O(n^2 \cdot \log q)$)

In case $g = 1$, we have a solution
of $1 = sa + tb$ in $\mathbb{Z}$
or of $1 = tb$ in $\mathbb{Z}_a$,
whereas in case $g > 1$, (or $\deg g \neq 0$)
there does not exist a solution of
either equation.

First primitive:

public key encryption scheme

Example: RSA (Rivest, Shamir & Adleman 1978)

Situation

public key

private key

$$\text{Alice} \xrightarrow{\text{secret channel?}} \text{Bob}$$

↑

Eve

$$y = \mathrm{enc}_{\text{public}}(\overset{x}{\overset{\shortparallel}{\text{msg}}}) \longrightarrow z = \mathrm{dec}_{\text{private}}(y)$$

Hopefully: $z = x$.

$\boxed{\text{Correctness}}$

All operations generating key pair, encryption, decryption must be 'efficient'....

(a) polynomial time (in the asymptotic setting)

(b) within 'seconds' (in the real world)

$\boxed{\text{Efficiency}}$

SECURITY

Eve should be unable to decrypt* knowing only the public key and the encrypted message (and maybe 'some' encryptions of plaintexts chosen by her).

* even partially

RSA key generation

Input: a security parameter n

Output: a public key (N, e) and a private key (N, d)

1. Generate a $n/2$-bit prime p.
2. Generate a $n/2$-bit prime q.
3. $N \leftarrow p \cdot q$.
4. $L \leftarrow (p-1)(q-1)$ "repetition length" $L = \varphi(N)$
5. Choose e, d, $0 < e, d < L$, such that $e \cdot d = 1$ in $\mathbb{Z}_L$. (Using EEA!)
6. return $\underbrace{(N, e)}_{\text{public key for encryption}}$, $\underbrace{(N, d)}_{\text{private key for decryption}}$

RSA encryption

Input: (N, e) public key, message $x \in \mathbb{Z}_N$.

Output: $y \in \mathbb{Z}_N$.

1. $y \leftarrow x^e$ in $\mathbb{Z}_N$.
2. return y.

RSA decryption

Input: (N, d) private key, ciphertext $y \in \mathbb{Z}_N$

Output: $z \in \mathbb{Z}_N$.

1. $z \leftarrow y^d$ in $\mathbb{Z}_N$.
2. return z.

Correctness?

$$z = y^d \qquad y = x^e \quad \text{in } \mathbb{Z}_N$$

$$= x^{ed} \qquad\qquad ed = 1 \text{ in } \mathbb{Z}_L \text{, ie.}$$

$$\qquad\qquad e \cdot d = 1 + k \cdot L \text{ in } \mathbb{Z}$$

$$= x^{1+k\cdot L}$$

$$= x$$

because we have $x^L = 1$
in case x is invertible in $\mathbb{Z}_N$.
by the Theorem of Lagrange,
or Euler.

<u>Thm</u> (Lagrange)

Given a group $(G, \cdot)$, finite, (commutative,)
we have $x^{\#G} = 1$ in G
for any $x \in G$. $\triangleleft$

We use $G = \mathbb{Z}_N^\times$ unit group of
the ring $\mathbb{Z}_N$ of
integers modulo N.

Its elements are all invertible elements of $\mathbb{Z}_N$.
Its operation is the multiplication inherited from $\mathbb{Z}_N$

⌜ P : given a, b invertible in $\mathbb{Z}_N$ check that $a \cdot b$ is invertible.
A ✓
N : check that $1 \in \mathbb{Z}_N^\times$
I : given a invertible, take b % $ab = 1$, check that b is invertible.
C ✓ ⌟

unit := an invertible element (in a ring).

cled[2]
21.11.07
5

Thus we know by Lagrange the ~~that~~

Theorem of Euler

Given $N \geq 2$, $x \in \mathbb{Z}_N^\times$,
we have
$$x^{\varphi(N)} = 1$$
where
$$\varphi(N) := \# \mathbb{Z}_N^\times . \qquad \square$$

It remains to compute $\# \mathbb{Z}_N^\times$ for $N = p \cdot q$.

Ad hoc:
Which elements are not invertible?

$$0, \; p, \; 2p, \; 3p, \; \ldots, \; (q-1)p$$
$$q, \; 2q, \; 3q, \; \ldots, \; (p-1)q$$

These are all different!
⌜ If $\alpha p = \beta q$ then because p, q are different primes: $q \mid \alpha$ and $p \mid \beta$ so $\alpha p = \beta q = 0$
thus $\alpha = 0 = \beta$. ↯ $\alpha, \beta > 0$. ... ⌟

And no other is non-invertible!
⌜ $\gcd(x, p \cdot q) \in \{1, p, q, p \cdot q\}$, $0 \leq x < N$.
If $g > 1$ then $g = p$ ie $x = \alpha p$
or $g = q$ ie $x = \beta q$
or $g = p \cdot q$ ie $x = 0$. ⌟

Thus
$$\# \mathbb{Z}_N^\times = N - 1 - (q-1) - (p-1)$$
$$= (p-1)(q-1) = L.$$

Thus we have

$$z = x$$

whenever x is invertible in $\mathbb{Z}_N$,
i.e. $\gcd(x, N) = 1$.

Actually, also if x is not invertible
one can prove $z = x$.

But there are only $p+q-1$ non-invertible elements
among $p \cdot q$ possible elements.

Standard: chance to pick a non-invertible element at random is $2^{-\text{size}}$.
That's practically zero.

Efficiency?

enc/dec : square & multiply in $\mathbb{Z}_N$ (repeated squaring)

$\longrightarrow \mathcal{O}(n^3)$

(this is poly-time and about a second for $n = 1024$ on a real computer)

key generation: $\mathcal{O}(n^4)$ for generating primes
everything else is cheaper...

elect² 21.11.07 ②

Security?

Complete breaks:

(1) Factor N, i.e. find p, q
(2) Find L.
(3) Find d.
(4) Find x.

given (N, e, y).

Actually: $(1) \Leftrightarrow (2) \Leftrightarrow (3) \Rightarrow (4)$

$$(T-p)(T-q) = T^2 - (N-L+1)T + N$$

Suppose you solve (3) twice: $e_1 d_1 = 1$ in $\mathbb{Z}_L$.
$e_2 d_2 = 1$

then $L \mid \gcd(\underbrace{e_1 d_1 - 1}, \underbrace{e_2 d_2 - 1})$

with high probability $\gcd = L$ or $2 \cdot L$ or $3 \cdot L$ ...

Case $bit_0(x) = 0$: $x = 2x'$

then $y = x^e = 2^e \cdot \underbrace{x'^e}_{=: y'}$

OPEN QUESTION: $(4) \Rightarrow (3)$?

Still, this is much less than wanted!

Another break:

(5) Find $bit_0(x)$ given (N, e, y), ...

Claim: An algorithm for (5) allows to construct an algorithm for (4).

" $(5) \Rightarrow (4)$ ":

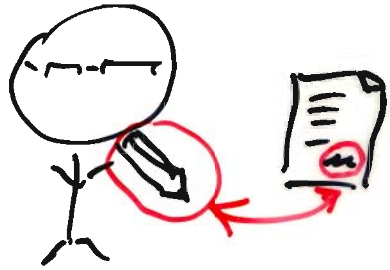
elect[2]
21.11.07
8

If we had (5) => (1)
then breaking RSA in that sense (5)
means that the attacker can
factor n-bit numbers.

But we assume that factoring
is difficult.

This would be "security reduction".

Signature?



It should

(1) identify signer

(2) "identify" the document (no changes)

(3) ~~chain~~ link signer and document

ElGamal signatures (and DSA or ECDSA)

DSA: digital signature algorithm

key generation:

Choose a large prime p (1024 bit)
and a large prime q (160 bit)
such $q \mid p-1$. — global setup

Choose $g \in \mathbb{Z}_p^\times$ such that

$$g^q = 1 \quad \text{and} \quad g \neq 1.$$

(To do so choose $h \in \mathbb{Z}_p^\times$ arbitrarily and compute $g = h^{\frac{p-1}{q}}$, until $g \neq 1$.)

Fix a hash function $\mathrm{hash}\colon \{0,1\}^* \to \mathbb{Z}_q$, $*\colon \mathbb{Z}_p^\times \to \mathbb{Z}_q$

? Choose a private key $\alpha \in (\mathbb{Z}_{p-1}, +)$

Compute the public key $a = g^\alpha$ in $(\mathbb{Z}_p^\times, \cdot)$

dlect[2] 28.11.07 ②

Signature verification:

Input: m message, $\sigma = (b, \gamma)$ signature, a public key of signer.

Output: Accept (1) or Reject (0)

0. verify $b \in \mathbb{Z}_p^\times$, $\gamma \in \cancel{\mathbb{Z}_{p-1}}\ \mathbb{Z}_q$.
1. Check

$$\boxed{a^{b^*} \cdot b^{\gamma} = g^{\mathrm{hash}(m)}} \quad \text{(key)}$$

public key

signature

where $* : \mathbb{Z}_p^\times \longrightarrow \mathbb{Z}_q$ has almost no structure and is <u>very</u> easy to compute (and is almost surjective)

for example if $\hat{b} \in \mathbb{Z}$, $0 < \hat{b} < p$ then $*(\underbrace{\hat{b} \bmod p}_{=b}) = \hat{b} \bmod q$.

How could we solve the key equation for (b, γ)?

First try: choose $b \in \mathbb{Z}_p^\times$ and solve for γ:

$$b^{\gamma ?} = g^{\mathrm{hash}(m)} a^{-b^*} = g^{\mathrm{hash}(m) - \alpha b^*}$$

This is a — so-called — discrete logarithm problem in our case in $\mathbb{Z}_p^\times$, or in $\langle g \rangle$. $\langle g \rangle = \{1, g, g^2, \dots\}$

Second try: Choose $\gamma \in \mathbb{Z}_q$ and solve the key equation for b:

elect² 28.11.07 3

$$a^{b^*} b^{\gamma} = g^{\mathrm{hash}(m)}$$

That looks even weird.

Nobody yet, had any ideas how to solve that...

→ brute force!? → time $\mathcal{O}(p)$ ≈ $\mathcal{O}(2^{1024})$

→ birthday? + clever if possible: time $\mathcal{O}(\sqrt{q})$ at best

Still $\sqrt{q} \sim 2^{80}$.

Third try: Try to solve for b, γ simultaneously...

Even less ideas...

How to solve with extra knowledge?

Note that $a = g^{\alpha}$ ← known to signer!

Look at the key equation

$$a^{b^*} b^{\gamma} = g^{\mathrm{hash}(m)}$$

power of g

? ← Make it a power of g!

clect[2]
28.11.07
4

Signature generation:

Input: global setup,

α private key,

m message.

Output: $\sigma = (b, \gamma)$ a (valid) signature

1. Choose $\beta \in \mathbb{Z}_q$ and compute $b = g^\beta$ in $\mathbb{Z}_p^\times$.

2. Solve the key equation:

$$g^{\alpha b^* + \beta \gamma} = g^{hash(m)} \text{ in } \mathbb{Z}_p^\times$$

for γ, or rather equivalently

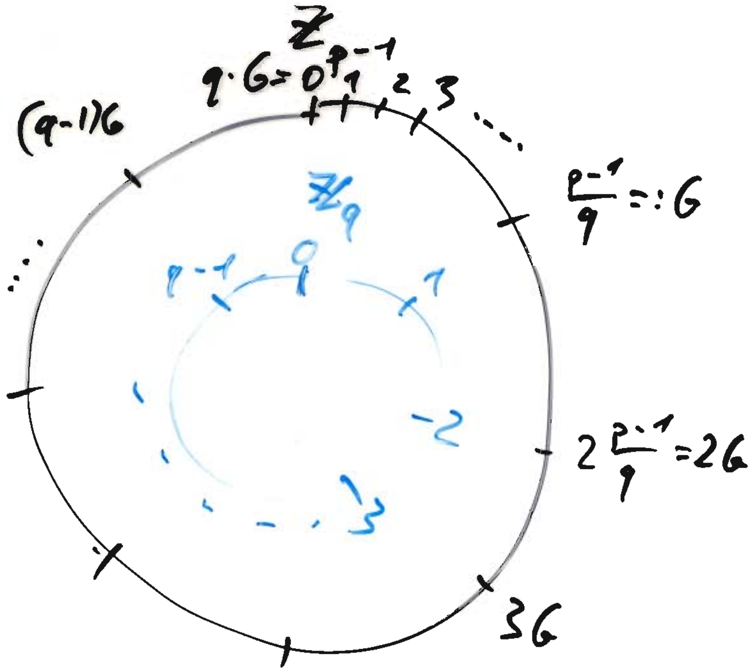
$$\boxed{\alpha b^* + \beta \gamma = hash(m) \text{ in } \mathbb{Z}_q}$$

ie. $\gamma = \beta^{-1} (hash(m) - \alpha b^*)$ in $\mathbb{Z}_q$

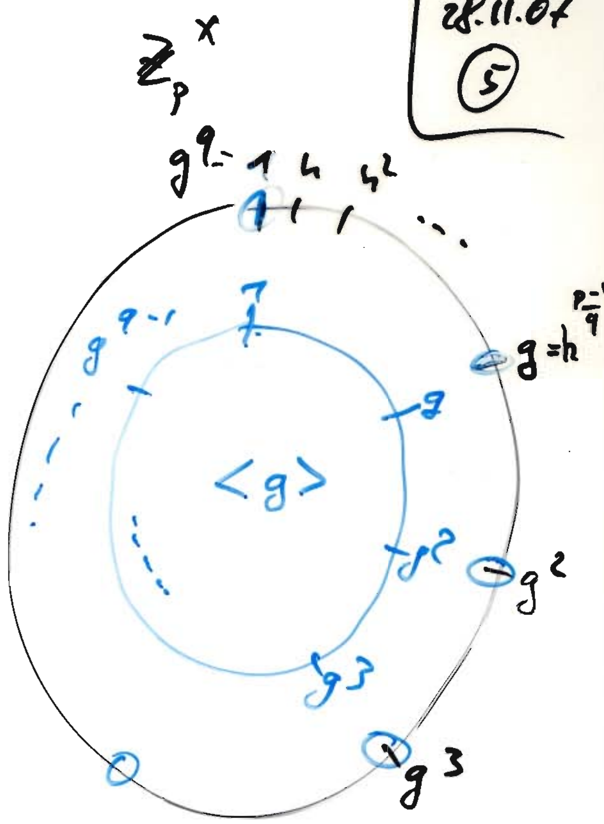
by EEA

3. Output (b, γ).

A little background



exponent group
(for $\mathbb{Z}_p^\times \ni h$)



'public' group

By the Lagrange theorem (or Euler or little Fermat) we know that $h^{p-1} = 1$ for every $h \in \mathbb{Z}_p^\times$.

(since $\# \mathbb{Z}_p^\times = p-1$ when p is prime).

Thus $g = h^{\frac{p-1}{q}}$ fulfills $g^q = 1$.

Actually then $g \neq 1$ implies that $g^k \neq 1$ for all $0 < k < q$.

⌜ By EEA get s, t such that $k \cdot s + q \cdot t = 1$.
(Note that q is prime!)
So $1 \neq g = g^{k \cdot s + q \cdot t} = (g^k)^s (g^q)^t = (g^k)^s = 1$
if we had $g^k = 1$. So $g^k \neq 1$. ⌟

Now we consider the map

secret world $\qquad\qquad\qquad\qquad$ public world

$$\exp_g : \begin{array}{ccc} \mathbb{Z}_q & \longrightarrow & \langle g \rangle \subset \mathbb{Z}_p^\times \\ \alpha & \longmapsto & g^\alpha \end{array}$$

(This is: $\begin{array}{ccc} \mathbb{Z}_{p-1} & \longrightarrow & \mathbb{Z}_p^\times \\ \beta & \longmapsto & h^\beta \end{array}$

after first multiplying with $\frac{p-1}{q}$: $\beta = \alpha \cdot \frac{p-1}{q}$.)

By the previous this map is (well-defined &) injective.

The image are precisely the powers of g, which form a subgroup $\langle g \rangle$ of $\mathbb{Z}_p^\times$.

$$\langle g \rangle = \{1, g, g^2, \ldots, g^{q-1}\}$$

Computing values of the map $\exp_g$ is easy: square & multiply! $\to O(n^3)$

But: the inverse map, the discrete logarithm

$$\begin{array}{ccc} \mathbb{Z}_q & \longleftarrow & \langle g \rangle \\ \alpha & \longmapsfrom & g^\alpha \end{array}$$

is difficult hopefully!

DLP (discrete logarithm problem)

Given $x \in \langle g \rangle$ (in some group G)

find $\xi \in \mathbb{Z}_{\# \langle g \rangle}$

such that $x = g^{\xi}$.

Standard assumption are:

- the DLP for $\mathbb{Z}_p^\times$ with p prime is difficult (theory: no poly-time
  practice: p at least 1024 bit)

- the DLP in an elliptic curve group E ~~is~~ is difficult (theory: no poly-time
  practice: size of coordinates about 160 bit)

No proofs!

Exercise priority: 2.2, 2.5, 2.6.
Rest ((2.1), 2.3, 2.4) will be bonus.

Does the ElGamal scheme fulfill our requirements?

Correct? Yes, by construction: a signature solves

$$\boxed{a^{b^*} b^{\gamma} = g^{hash(m)}}$$

which is equivalent(!) to

$$\boxed{\alpha b^* + \beta \gamma = hash(m)}$$

Then (b, γ) is a signature to m.

Efficiency? Yes, only prime generation, group operation, modular computations (EEA), square & multiply.

Security?
What's the task of an attacker?

Gets: a, global setup: $\mathbb{Z}_p^\times$, g, hash, *, ... ξ,
$q = \# \langle g \rangle$.

Additionally, the attacker may read or even choose messages that'll be signed by α or others.

Task: Output a message with a valid signature, which was never queried.

Security Goal

poly-time

For example: what does non-existence of such a only poly-time restricted attacker imply for the hash function?

hash: $\{0,1\}^* \longrightarrow \mathbb{Z}_q$ in our scheme.

If there exists a poly-time algorithm that outputs $m_1, m_2 \in \{0,1\}^*$ such that $m_1 \neq m_2$,

$$\text{hash}(m_1) = \text{hash}(m_2)$$

} not collision-resistant

(Note: (m_1, m_2) always exist, but are possibly difficult to find.)

An attacker could using such a pair ask for a signature of m_2 and return m_1 with this signature. That contradicts security requirement.

Conclusion:

hash not collision-resistant } $\Rightarrow$ ElGamal with hash is broken.

or:

ElGamal using hash is secure } $\Rightarrow$ hash is collision-resistant.

elect²
3.12.07
3

1. identify signer? (not knowing α)

~~Sn~~ Somebody else cannot produce a valid signature because he would be an attacker in the sense of our security goal. So a valid signature identifies the signer.

2. identify document, prevent changes?

Producing a new (even if only slightly changed) document with valid signature from a given document again contradicts our security goal.

3. link document and signer?

... again contradicts our security goal:

It means to produce a document with valid signature for another person than the original signer.

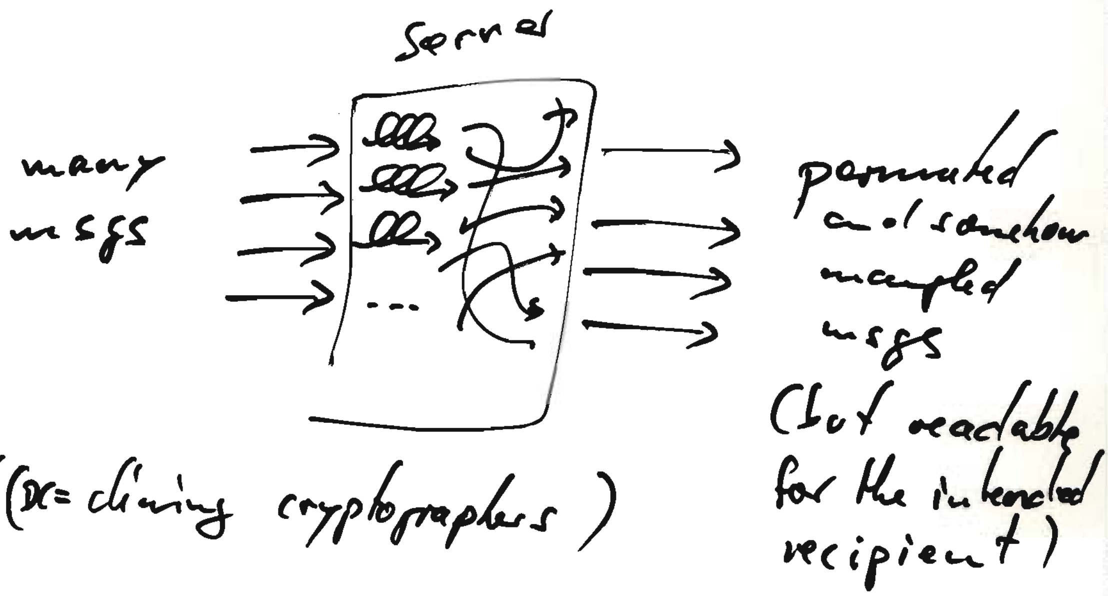
Crypto-primitives cont'd

elect² 3.12.07 4

1a Secure channels
→ Combine encryption and signatures.

1b Untappable channels
→ Physical security.

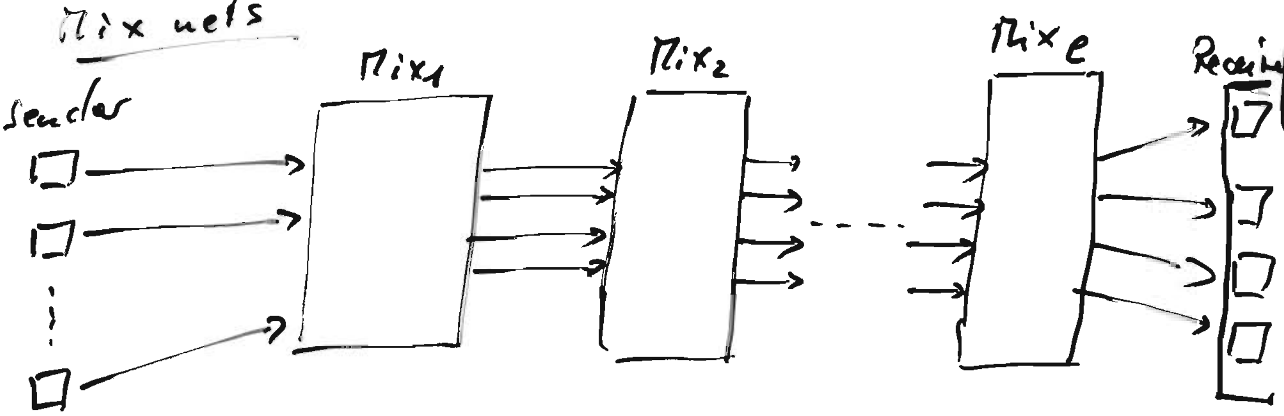
2 Anonymous channel
→ Mixnets



→ DC-net (DC = dining cryptographers)

elect²
12.12.07
1

Mix nets



Decryption mixnet

Sender i

Input: m_i, global setup.

Output: c_i.

1. Choose random string r_i.
2. Encrypt $m_i \| r_i$ with the public key of $\mathrm{Mix}_e, \mathrm{Mix}_{e-1}, \ldots, \mathrm{Mix}_1$:

$$c_i = E_1(E_2(\ldots\; E_e(m_i \| r_i)\ldots))$$

3. Return $c_i =: c_i^{(0)}$

Mix_j

Input: a list of messages $\left(c_i^{(j-1)}\right)_i$

Output: a list of messages $\left(c_i^{(j)}\right)_i$

1. Decrypt all messages with its private key:

$$\gamma_i^{(j)} = D_j\left(c_i^{(j-1)}\right).$$

2. Sort the new list $\left(c_i^{(j)}\right) = \mathrm{sort}\left(\gamma_i^{(j)}\right)$.
3. Return $\left(c_i^{(j)}\right)_i$.

Receiver: gets $(m_i \cancel{\| r_i})_i$.

Effect:

- No relation between output message position and sent message position.
  → Randomness important! Otherwise reencrypting reveals the sender.

So the messages do not reveal the sender and thus we have <u>anonymity</u>.

Problems:

- Can a corrupt Mix or several recover the connection message ⟷ sender?

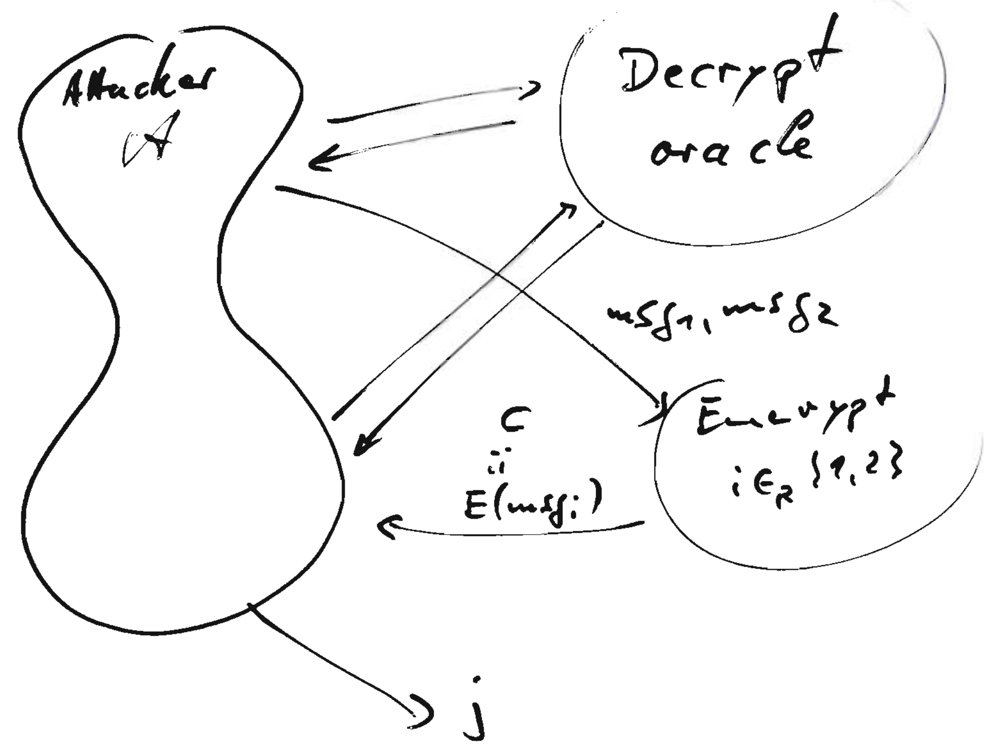
NO, if encryption is $\underbrace{\text{INDISTINGUISHABLE}}_{\text{Security goal}}$.

<u>Security goal</u>

There is no poly-time attacker that can ~~solve~~ win the following game:

Input: setup.
Intermediate output: msg_1, msg_2 of same length.
Intermediate input: $\underbrace{\text{Encryption of } msg_i}_{\text{challenge}}$ with $i \in_R \{1,2\}$.
Output: a bit j.
Additionally, he may query a decryption oracle.

He wins if $i = j$ and he never queried the challenge.



RSA itself does not fulfill that:

$\mathcal{A}$ could compute $E(msg_1)$ and $E(msg_2)$ and compare to c.

So the encryption must be randomized.

ElGamal encryption does it!

Setup: a group G and a generator g of it, $q = \text{order}(g)$ known with large prime factor or prime itself.

Encrypt: Input: $x \in G$ message, $a = g^\alpha$ public key of recipient.

Output:

1. Choose $s \in_R \mathbb{Z}_q$.
2. Return $(g^s, a^s \cdot x)$

Decrypt: Input: $(w, y) = (g^s, a^s x)$, α private key.

Output: x

1. Return $y/w^\alpha = a^s x/(g^s)^\alpha = g^{\alpha s} x / g^{s\alpha} = x$.

So if at least one Mix remains uncorrupted the entire process stays <u>anonymous</u>.

Danger: ~~the~~ mix net does <u>not</u> provide confidentiality or integrity.

Clear: it cannot give authenticity.

We might of course combine other means with a mix net to achieve these!

But be careful not to violate already granted properties.

Re-encryption mixnet

Sender: Inputs m_i
Output: $(g^{s_i}, a^{s_i} x)$

where $a = g^{\alpha}$ is the public key of the receiver.

Mix_j: Input: list of ciphertexts $(g^{s_{i,j-1}}, a^{s_{i,j-1}} x_i)$
Output: — " — $(g^{s_{ij}}, a^{s_{ij}} x_i)$ sorted...

1. Re-encrypt:
   choose $\delta_{ij} = s_{ij} - s_{i,j-1} \in_R \mathbb{Z}_q$
   calc $g^{\delta_{ij}}$ and $a^{\delta_{ij}}$
   and $(g^{s_{ij}}, a^{s_{ij}} x_i) = (g^{\delta_{ij}} \cdot g^{s_{i,j-1}}, a^{\delta_{ij}} \cdot a^{s_{i,j-1}} x)$
2. Sort.
3. Return list.

positive feature! Extra randomness!

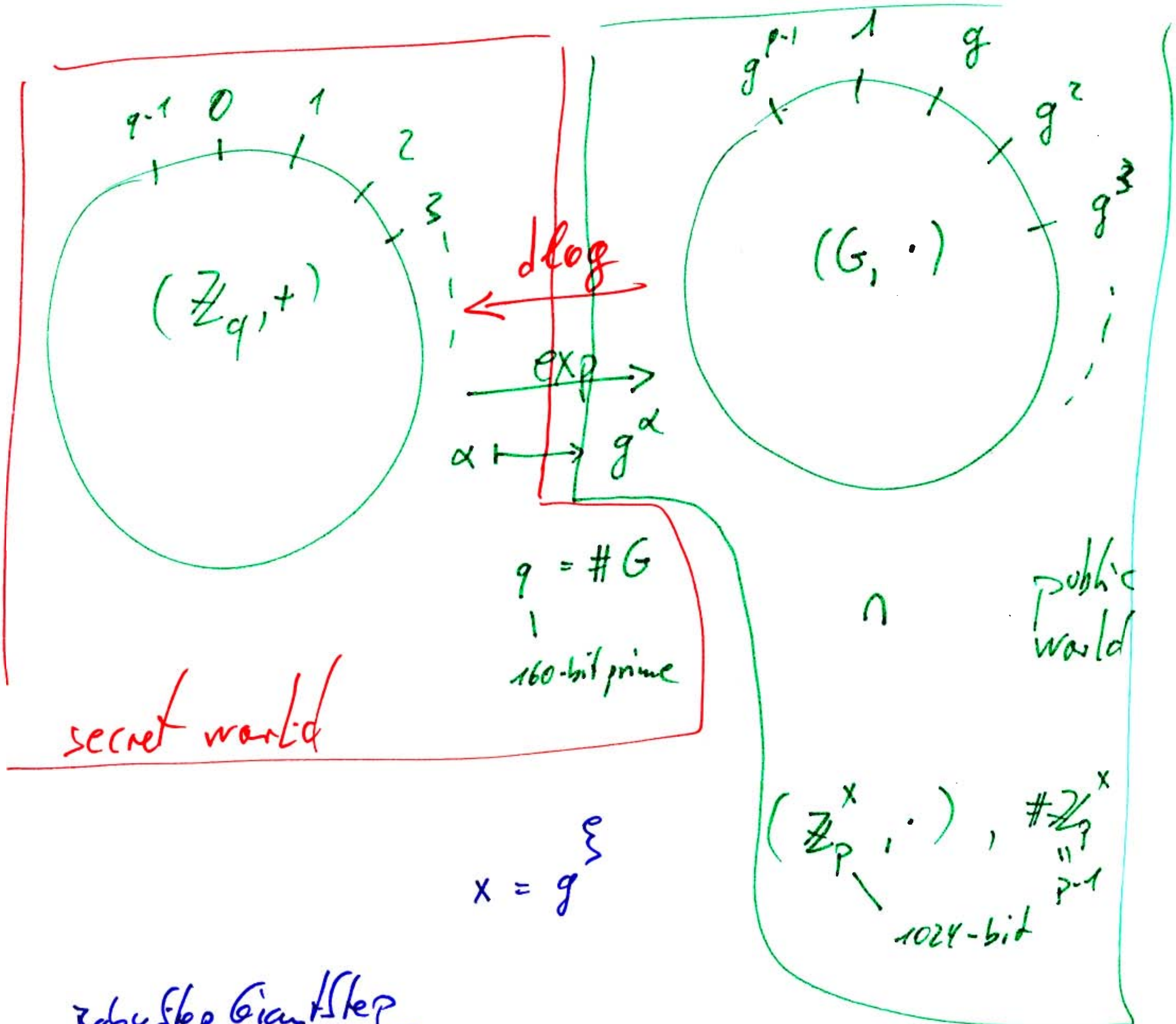
Pro & Cons

+ Sender has only one encryption to do.

+ list of mixes may vary, need not be known in advance
  → more robust, a non-working mix can be skipped or replaced.

- sender has no guarantee that a given list of mixes are used

+ same anonymity features even
+ better —"— because of new randomness in each stage.



$$x = g^\xi$$

Baby Step Giant Step

$$g^{-\xi_0} x = g^{\xi_1 \lceil \sqrt{q} \rceil}$$

$\rightarrow \quad O(\sqrt{q})$

Pohlig - Hellman $\quad q = p_1 \cdot r_1$ (small prime)

$$x^{r_1} = (g^{r_1})^{\xi_0}$$

$$\xi = \xi_0 + \xi_1 p_1 + \xi_2 p_1^2$$

$\rightarrow$ Determine ξ_0

Thm (Lagrange)

G finite group.
Then
$$g^{\#G} = 1$$
for $g \in G$.

17.12.07
Tutorial 8

ElGamal signatures

Verify: $a^{b^*} b^{\gamma} = g^{\mathrm{hash}(m)}$ in $\underline{G}$

a: public key of signer

$\underbrace{b^{\gamma}}_{\text{signature } (b,\gamma)}$

g: part of global setup

hash(m): message

Sign?

$a = g^{\alpha}$

$m, (b^*, \gamma)$: $\underline{b^* = \left(g^{\mathrm{hash}(m)} a^{-b^*}\right)^{1/\gamma}}$

Sign? $b = g^{\beta}$

$\alpha b^* + \beta \gamma = \mathrm{hash}(m)$

Solve and answer: (b, γ).

Global Setup: Choose $G \ni g$ of known order q ....
Choose (fix) hash, *.

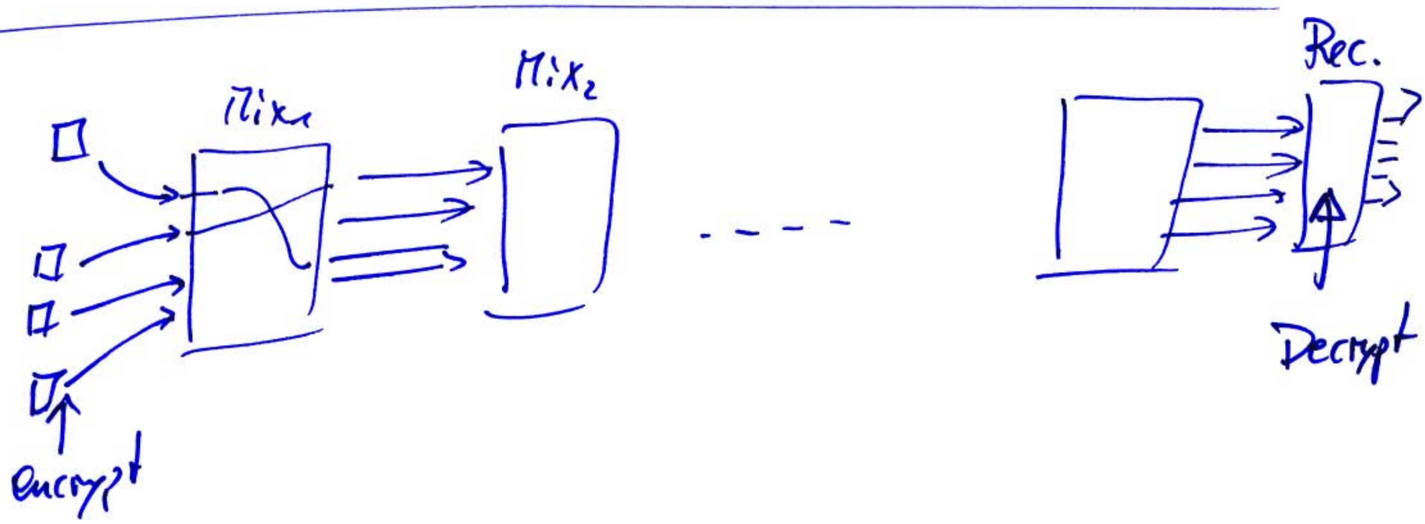
User Setup: Choose private key α.
compute public key $a = g^{\alpha}$.

17.12.07
Tutorial ③

ElGamal encryption

Sender		Recipient
$x \in \mathbb{Z}_p^{\times}$	$\xleftarrow{\quad a \quad}$	$a = g^{\alpha}$.
choose $\rho \in \mathbb{Z}_q$.	
	$\xrightarrow{(g^{\rho},\ a^{\rho} x) = (r, y)}$
		$z = \dfrac{y}{r^{\alpha}}$

$$z = y\, r^{-\alpha} = \underbrace{g^{\alpha\rho}}_{a^{\rho}} \times \underbrace{g^{-\rho\alpha}}_{r^{-\alpha}} = x \quad \checkmark$$



$$(g^{\rho}, a^{\rho} x) \longmapsto (g^{\rho'}, a^{\rho'} x)$$

$\cdot g^{\delta}$ $\cdot a^{\delta}$

17.12.07
Tutorial (4)

Given encryption scheme

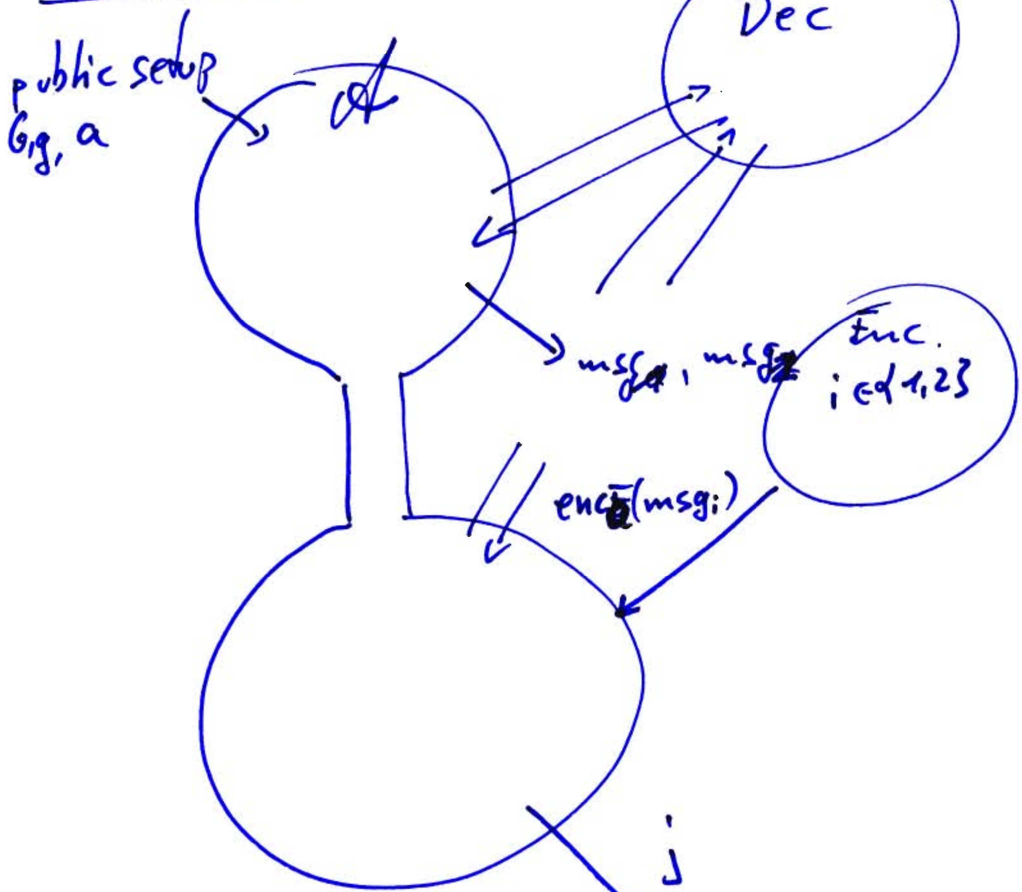
$$\text{setup} \to G, g \qquad \text{setup} \to a, \alpha$$

enc_a

dec_α

IND – CCA

- Indistinguishability
- Chosen Cipher text attack

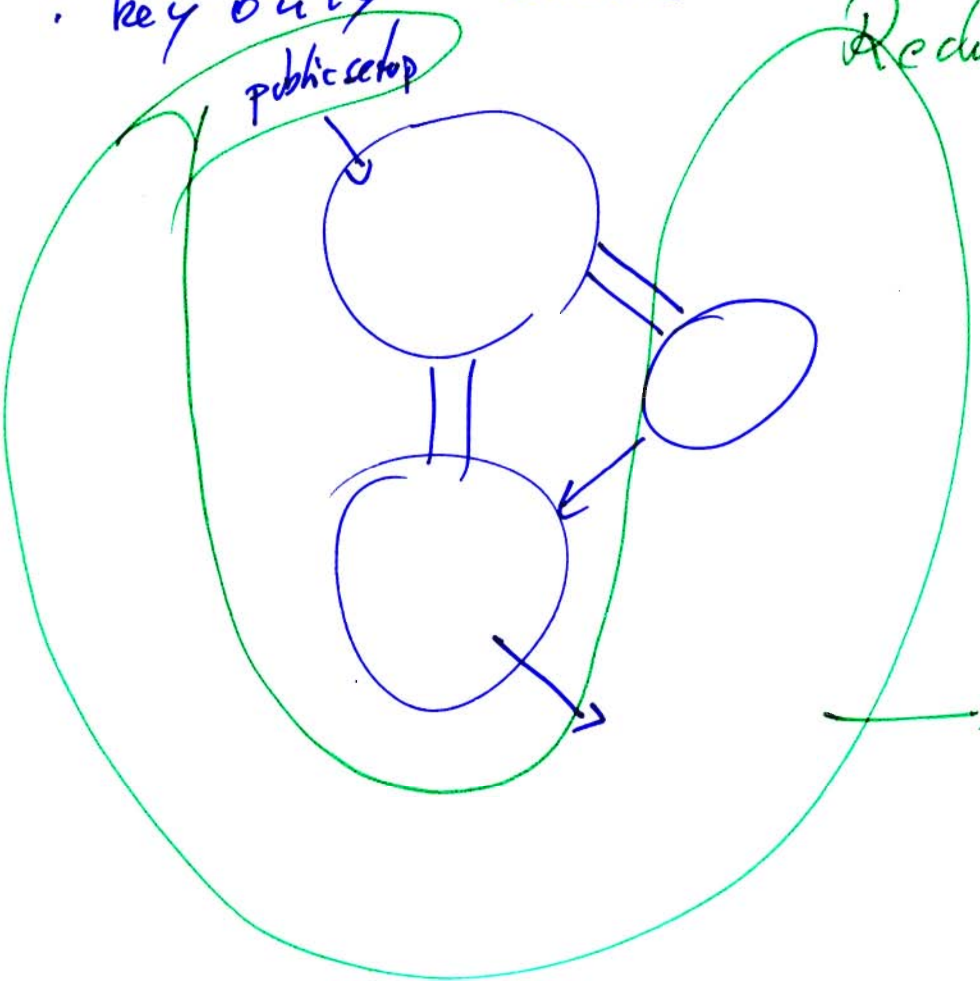


Win $\Leftrightarrow$ $i = j$?

RSA does not have this!

17.12.07
Tutorial (5)

- Indistinguishability
- key only attack



Reduction

$(g, g^\alpha, g^\beta, g^\gamma)$

$\gamma = \alpha\beta$?

skip remaining building blocks

Classification (rough):

Hidden voter	Hidden vote
=	=
anonymous submission of vote	encrypted submission of vote

$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}$
Hidden voter with hidden vote

Hidden voter

Scheme in Chaum (1981)

Announcement stage

- Chaum's decryption mixnet and its RSA public parameters: $E_K(m, r) = E_{K_1}(E_{K_2}(\ldots E_{K_\ell}(m \| r)\ldots))$
- Each voter is associated with a digital signature.

Registration stage

(1) Token generation: The eligible voter V_j generates a random RSA key pair K_{V_j} (public key) and $K_{V_j}^{-1}$ (private key) and set $token_j = K_{V_j}$.

2 V_j sends the $token_j$ in encrypted form to Mix_1 as

$$E_K(\, token_j \,,\, r_j \,) \quad (*)$$

and a digital signature on (*) to prove eligibility.
Mix_1 sends a receipt to V_j.
and process (*) through the mixnet.

3 Mix_e outputs a lexicographically ordered list of voter tokens $(token_{\pi(j)})_j$.
to a bulletin board. some permutation

Verification stage
Voter V_j verifies that $token_j$ is received and recorded correctly.

Voting stage
Voter V_j encrypts her vote v_j as new random string

$$E_K(\, \underbrace{token_j \,\|\, E_{K_j^{-1}}(v_j \,\|\, 0^k)} \,,\, r_j' \,)$$

and then sends this together with a signature to Mix_1 who acknowledges this with a receipt.

After mixing Mix_e outputs a lexicographically ordered list of $\underbrace{K_{V_j}}_{token_j} \| E_{K_{V_j}^{-1}}(v_j \| 0^k)$ on the bulletin board.

elect²
19.12.07
3

Eligibility

Only eligible voters can vote and cast 1 vote.

This is granted if Mix_1 not corrupted.

To guarantee that Mix_1 works correctly we could add that it has to prove that it only sent a single per ~~voter~~ eligible voter and no token for non-eligible ones.

Maybe a further bulletin board could do that.

This makes control of

(*) one eligible voter $\leftrightarrow$ one token

possible for everybody.

Also if a voter claims that her token has not arrived she can prove so by revealing her random string r_j and thus make that token invalid and later register a new one.

~~The~~

In the voting stage again the inputs to Mix_1 should be published and must be processed through the mixnet only after the election.

We need that the mixnet encryption uses randomness in each step, i.e.

(*) $$E_K(m, r) = E_{K_1}(E_{K_2}(\ldots E_{K_\ell}(m \| r_\ell) \ldots \| r_2) \| r_1)$$

so that no mix can reconstruct (*).

elect[2]
19.12.07
4

Anonymity

Granted if at least one Mix
remains uncorrupted.

Verifiability

Individual: everybody can check
that his own vote has
been correctly registered.

General: ✓

9.1.08

Receipt-freeness

NO!

→ Not fair, vote-selling and
family voting are possible.

Sako & Kilian (1995)

9.1.08

1

Announcement stage

Set up for a re-encryption mixnet:

a group G (e.g. $G \subset \mathbb{Z}_p^\times$, $\#G = q$, p, q prime,

$g \in G \setminus \{1\}$. [$\Rightarrow q \mid (p-1) = \#\mathbb{Z}_p^\times$])

a key pair (S_i, σ_i) with $S_i = g^{\sigma_i}$

for Mix_i, $1 \le i \le \ell$

and $K = \prod S_i = g^{\sum \sigma_i}$.

Registration stage

The eligible voter V_j registers and interacts with the mix net:

Input: $E_K(v^{(f)}, 0) = (g^0, K^0 v^{(f)})$

for $f \in$ SetOfCandidates.

For each voter V_j and for each mix Mix_i

1. Mix_i chooses a permutation π_{ij} and commits to it to voter V_j.
(i.e. give locked boxes with π_{ij} in them to voter V_j).

2. Re-encrypt with a random string r_{ij}

$$(g^{s'}, K^{s'} v^{(f)}) \longmapsto (g^{s}, K^{s} v^{(f)})$$

$s = s' + r_{ij}$

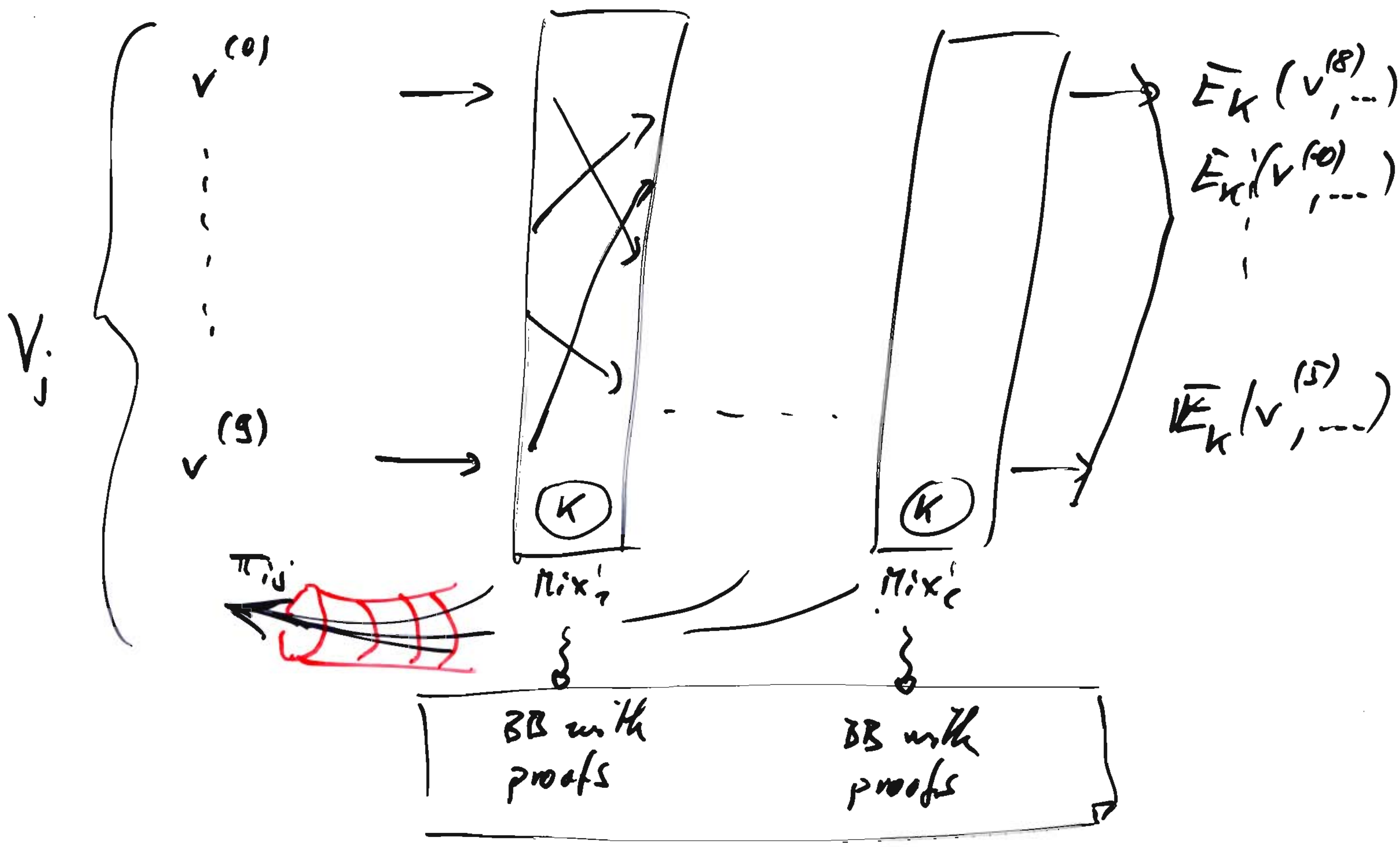
9.1.08
2

3. Permute these re-encrypted votes using π_{ij} (varying the f's).

4. Post a non-interactive proof of correct re-encryption and permutation on a bulletin board.

5. Decommit π_{ij} to voter V_j over an untappable channel, ~~that~~ who verifies π_{ij} using the posted proof.

Output: $\{ E_K(v^{(f)}, \sum_i r_{ij}) \mid f \in SetOfCandidates \}$



Voting stage

1. Vote casting

   V_j chooses one of the ElGamal encryptions from the output of the mixnet, and sends it.

2. Mixing

   After election day all encrypted votes are sent through the decryption mixnet and post non-interactive proofs of correct mixing

$E_K(v^{(f)}, r_{jf})$

$\|$

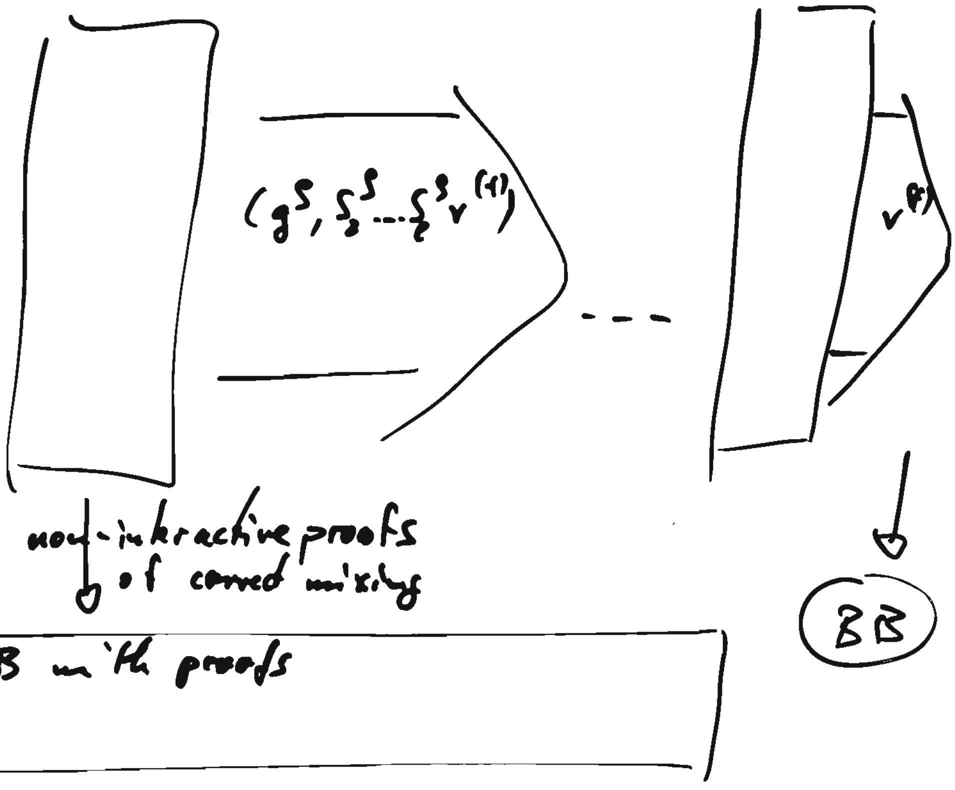
$(g^\rho, K^\rho v^{(f)})$

$=$

$(g^\rho, S_1^\rho S_2^\rho \cdots S_\ell^\rho v^{(f)})$

$K = \prod S_i$

$= g^{\sum s_i}$



Eligibility

As in Chaum (1981).

one voter <-> one vote!

Anonymity (privacy)

Given, relies on the untappable channel.
But then anonymity is even unconditional.
If you replace the untappable channel with a private channel (so using encryption and signatures) the anonymity is only computational.

Verifiability

Individual : ~~all~~ correct tokens ✓

~~all~~ own votes counted : ? Yes!
(Look at the proofs of correct mixing in the mixing in the voting stage).

General : ✓

~~Fairness~~

That means : no receipt proves the value of the vote.

Accuracy

Everything works correctly and provably so.

Fairness No one can compute a partial tally.

Scalability Problematic.

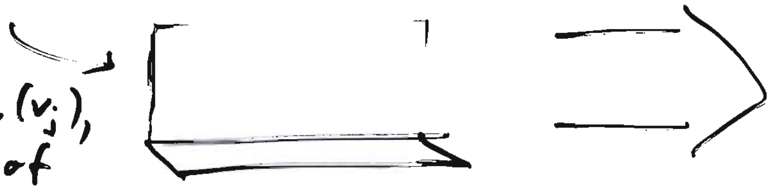
Robustness Decryption mixnet can be interrupt by each single mix.

16.1.08
elect²
1

Hidden vote

Voter 1

$E_K(v_j)$,
proof



Authority: First "multiplies" the encrypted votes and gets

$E_K(\sum v_j)$

then decrypt this!

Output $\sum v_j$

and proofs...

Benaloh & Fischer '85: single authority.

Benaloh & Yung '86: splitted authority.

Benaloh '87: .... + robustness (by using a threshold secret sharing)

Setup: $N = p \cdot q$, p, q primes, $p \neq q$,

r prime with $r \mid p-1$.

$K \in \mathbb{Z}_N^\times$ public key of the authority such that K is not an rth power.

→ rth residue: $K^r \neq 1$

→ rth power: $K \neq x^r$ for all x.

All elts. mod p have $x^{p-1} = 1$. Some do not have this for smaller exp. Then $(g^r)^{\frac{p-1}{r}} = 1$. So if K is an rth power then $K^{\frac{p-1}{r}} = 1$. And vice versa.

$$E_K(v_j, u_j) = K^{v_j} u_j^{r} \quad \text{in } \mathbb{Z}_N^\times.$$

By raising this to $\frac{p-1}{r}$ th power we obtain:

$$\left(K^{\frac{p-1}{r}}\right)^{v_j} \cdot 1 \quad \text{in } \mathbb{Z}_p^\times.$$

Prevoting stage (= registration stage):

Each voter submits a vote to 0 and
a vote to 1
and a interactive or non-interactive proof
that this is what is claimed.
towards an authority.

Voting stage
Each voter submits one of her proposals.

Tallying:
The authority multiplies:

$$K^{\sum v_j} \cdot \left(\prod u_j\right)^{r} \quad \text{in } \mathbb{Z}_N^\times$$

and then decrypts:

$$\left(K^{\frac{p-1}{r}}\right)^{\sum v_j} \quad \text{in } \mathbb{Z}_p^\times$$

and derive $\sum v_j$ from it... by computing
the discrete log using Babystep giantstep

⚠ exponential time but feasible with $< 2^{40}$ voters.
or similar

elect[2]
16.1.08
3

Eligibility

Anonymity As long as the authority is not corrupted, it's anonymous.
Therefore Benaloh & Yung '86 splited the authority.
But still that scheme is not robust.

+ Robustness So Benaloh '87 used a threshold secret sharing to make it more robust.
with a (t, k)-threshold scheme
k : #shares in total
t : #shares needed for secret
we have k authorities and as long as t of them work properly the decryption can be done.

Accuracy & Verifiability Both universal and individual by checking all the proofs.

Fairness Multiple decryption authorities grant that.

Scalability More or less yes.
(Only possible problematic point is the tallying.)

lect 2
16.1.08
4

Questions on schemes

~~Describ~~

- Produce a 'poster like' description of the scheme.

- Properties:
  - Eligibility
  - Anonymity / Privacy
  - Verifiability
  - Robustness
  - Scalability

- More properties...

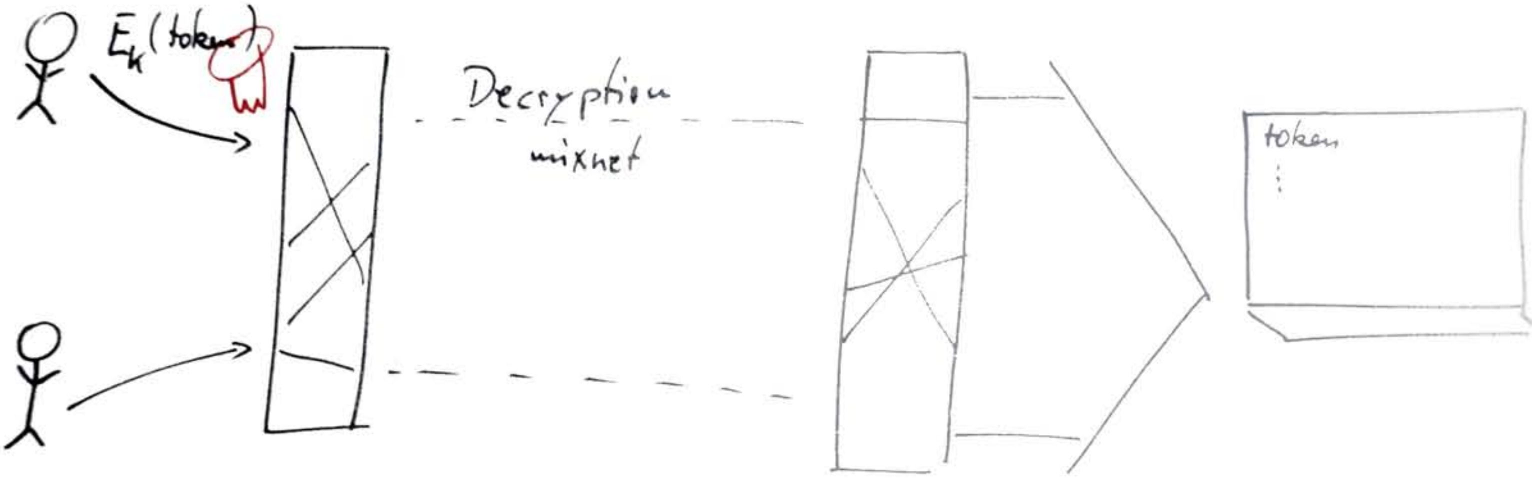
- Social impact and practical considerations.

Exam date: Friday 11 april '08
10.00

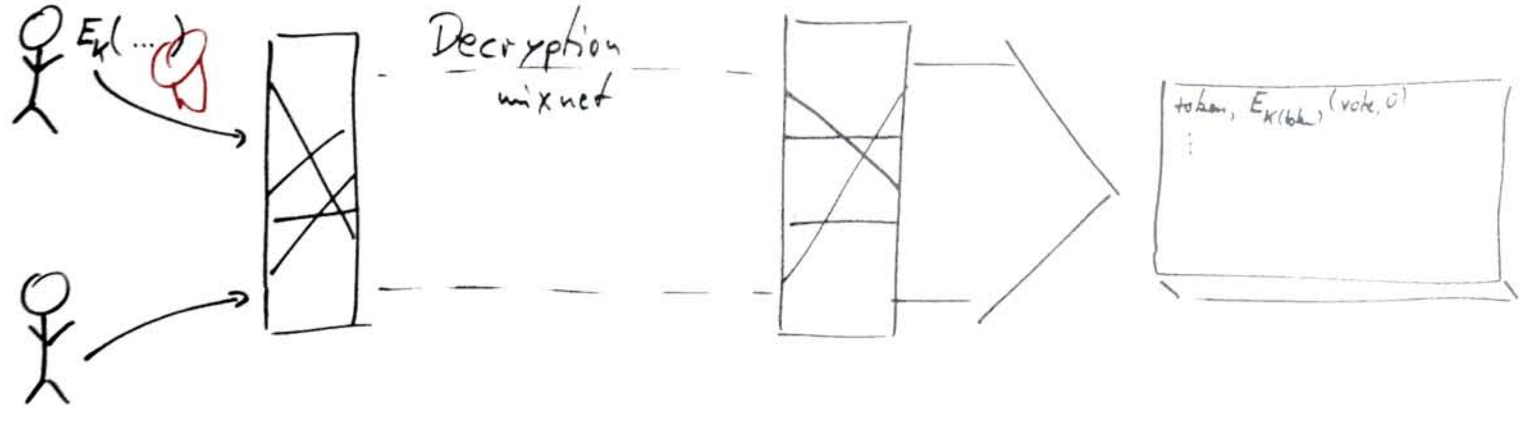
Chaum (1981)

Announce

Registration stage



Voting stage



$$E_K(\text{plaintext}) = E_{K_1}(\dots\ E_{K_3}(\text{plaintext}, r_3) \dots, r_1)$$

$$E_{K_1}((1,p), r) = (g^r, K_1^r p)$$

Chaum (1981)

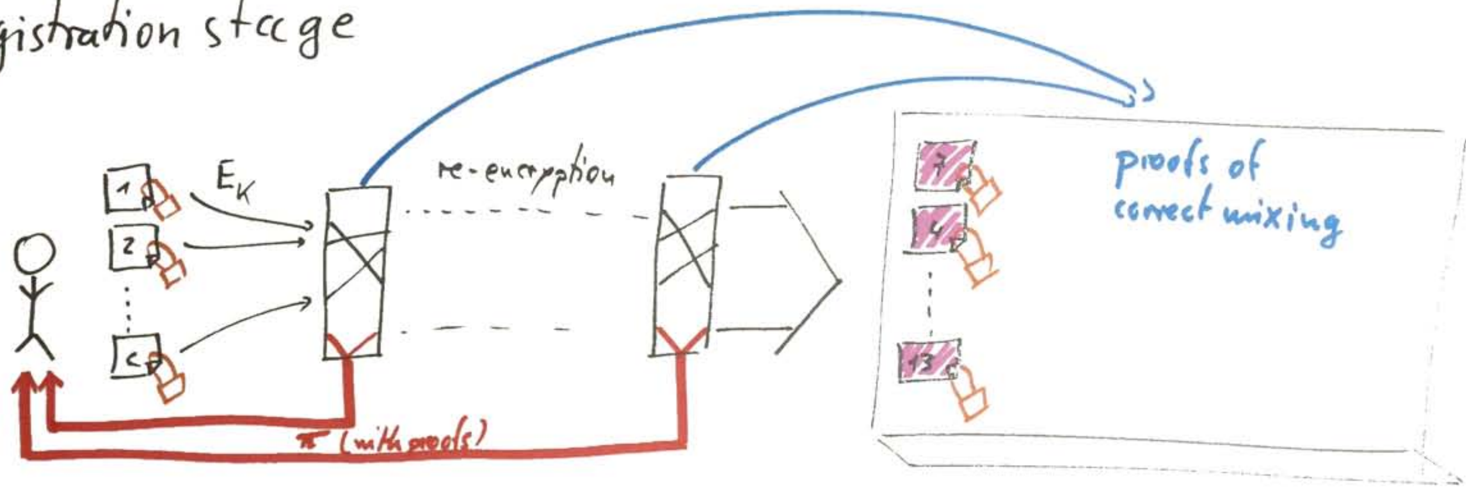
<u>Questions:</u>

- Scalability?
- Security?
- Where does the secret keys come from?
- How is the signing done?

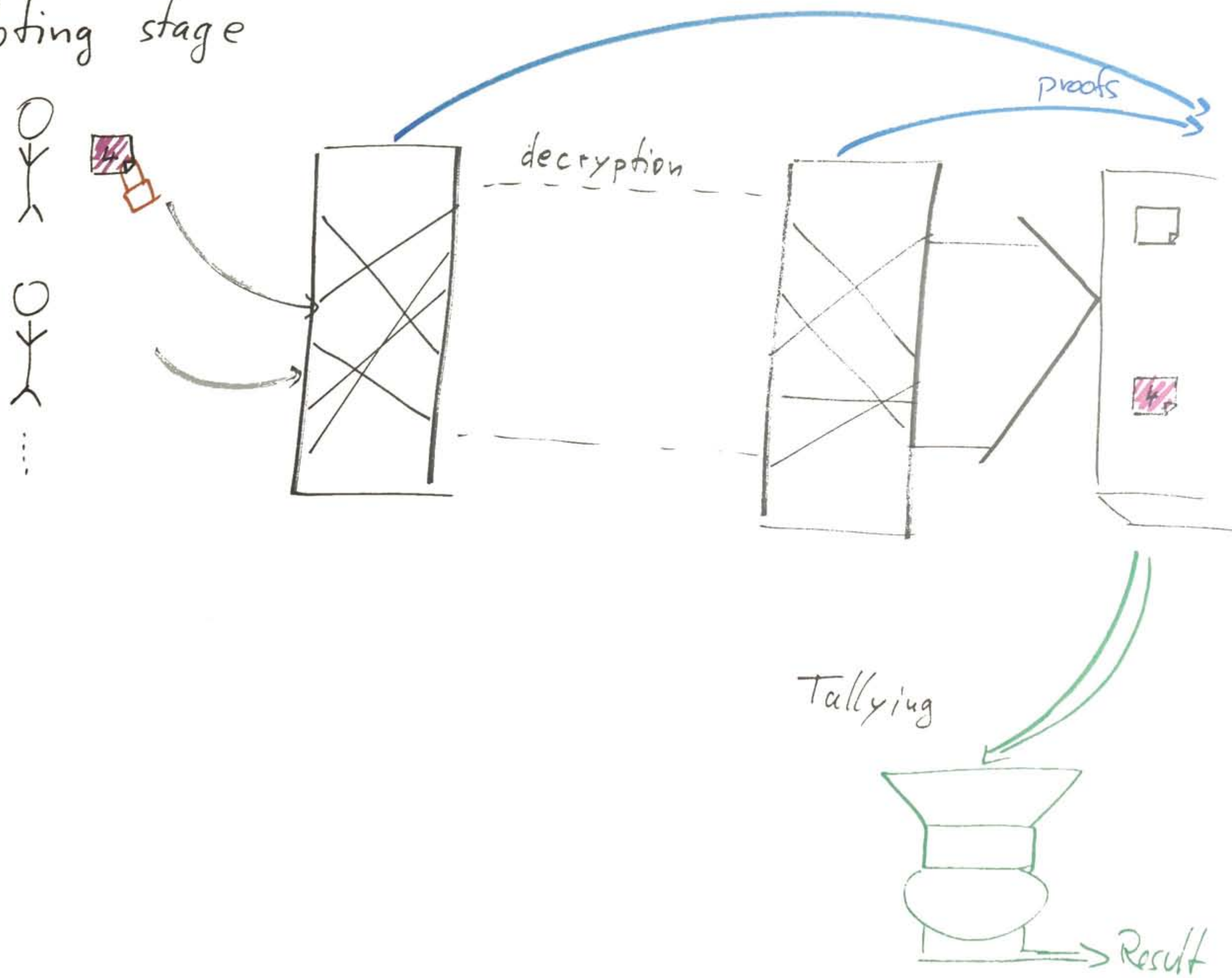
Sako & Kilian (1995)

Setup ...

Registration stage



Voting stage



Sako & Kilian (1995)

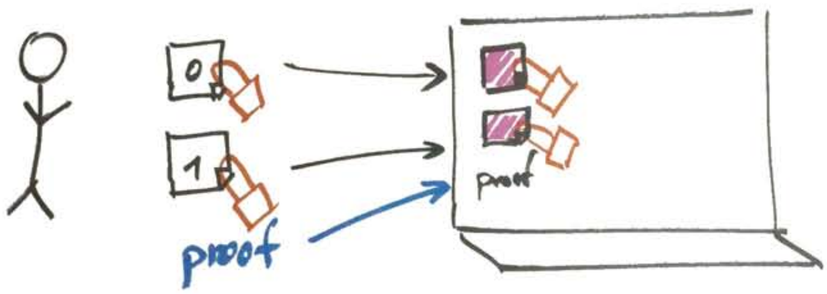
Questions:

- Why do we need the registration stage?
  ↳ because of receipt freeness!

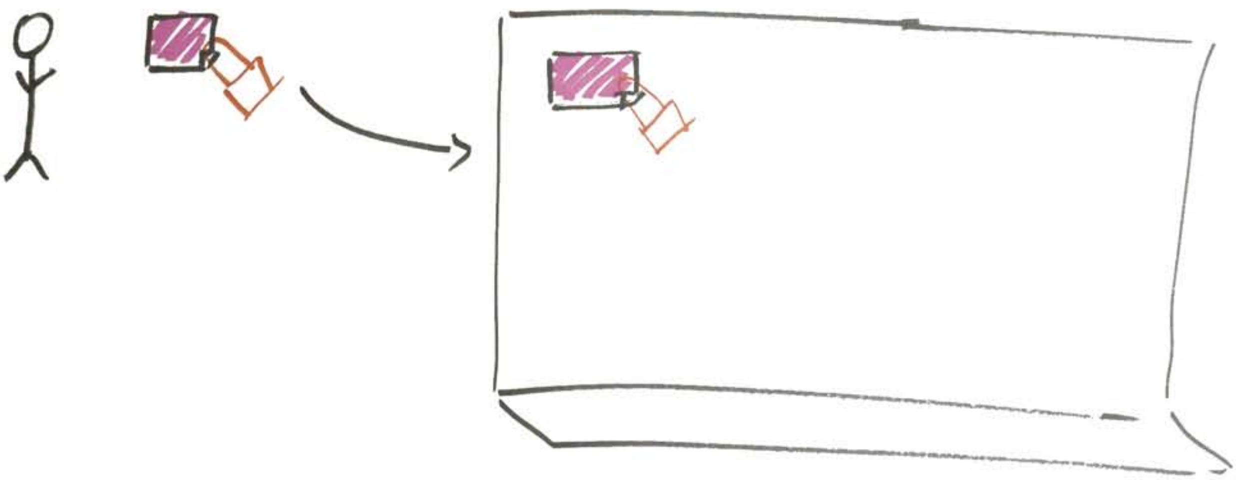
Benaloh & al (1985, 86, 87)

Setup: $N = p \cdot q, \quad r \mid (p-1).$

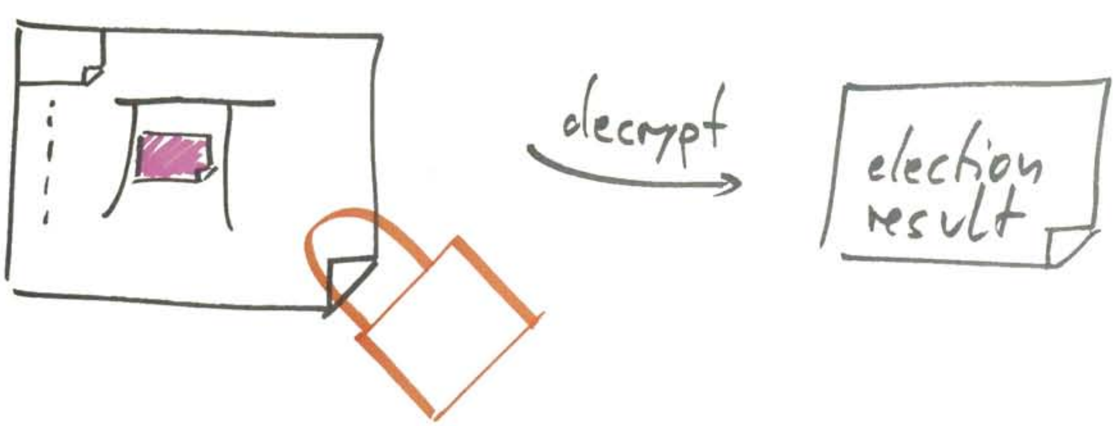
Prevoting stage



Voting stage



Tallying stage

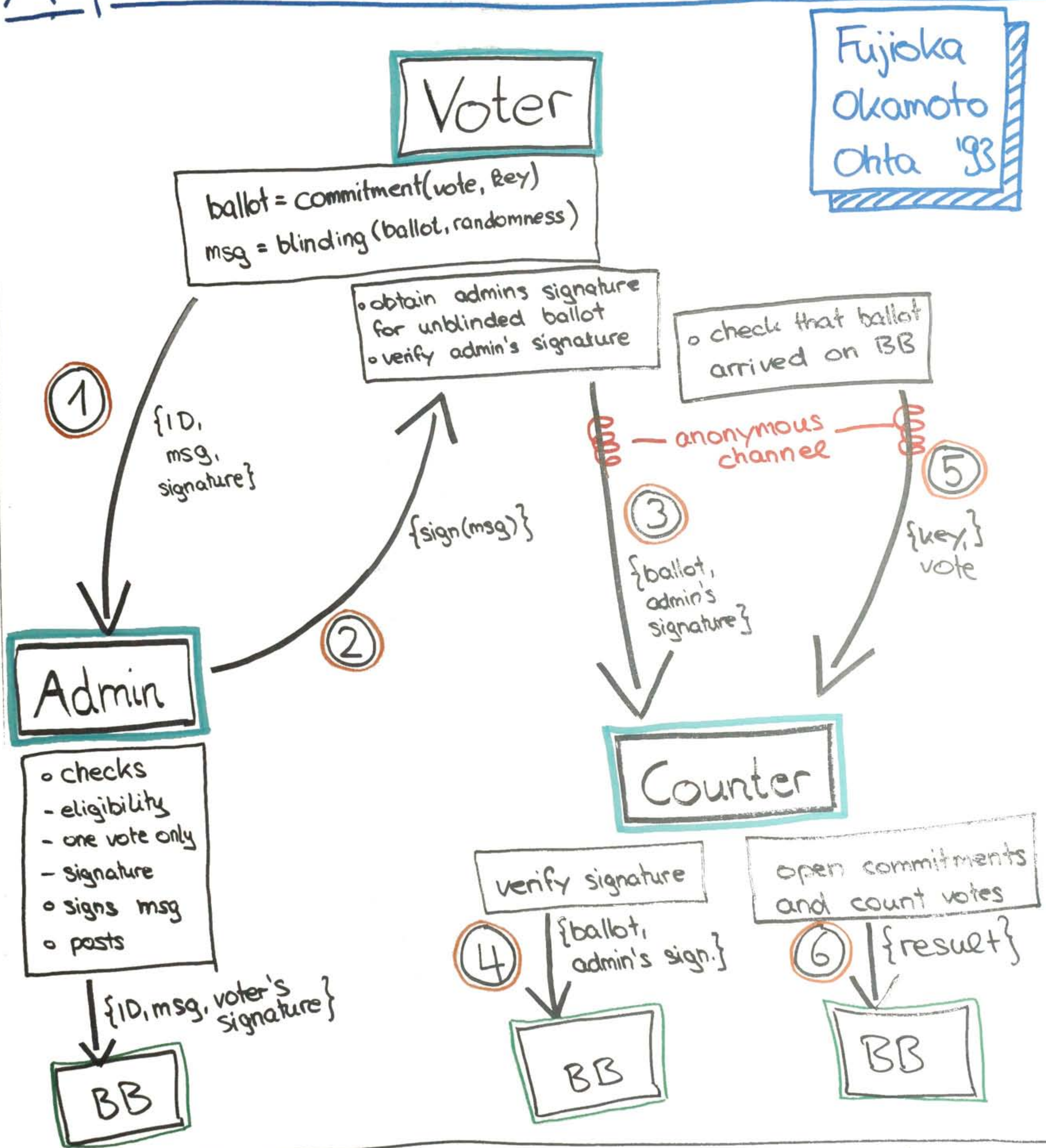


Benaloh & al (1985, 86, 87) Hidden vote

<u>Questions:</u>

- Eligibility?

A practical secret voting scheme for large scale elections



- Admin's signature invalid ⟶ reveal ballot & signature
- More/less votes than counted ⟶ reveal randomness
- Own ballot not on counter's list ⟶ reveal ballot & admin's signature

Fujioka & Co

- commitment ≠ encryption
  - ⇝ like a box with a lock that you give to someone
  - ⇝ you cannot change the contents
  - ⇝ the other person cannot see the contents without the key

Eligibility: yes, by the admin

Anonymity: - using blind signature scheme
- using an anonymous channel (MixNet)

Verifyability: individual and global because everything is on bulletin boards

Receipts: The voter has his ballot and his key so he can in principle prove what (s)he voted for :(

Robustness: yes, the admin can be distributed, as can be the list of voters.
Same for the counter...

- How to check that all votes were counted?
  - ⇝ Check the numbers of items on the bulletin boards
- Can somebody take a ballot & a key and claim they're his vote? No, because he doesn't know the randomness that was used to construct the blinded message which is on the admin's bulletin board...

⇝ There is some delay between collecting and opening because one needs the list index on the counter's board before sending the key and the vote doesn't arrive there immediately because the mixnet needs to collect enough messages first...

Kiayas & Yung (2002?)

- n Voters V_j, $j = 1, \dots, n$
- Bulletin Board Authority (BBA)

(1) $V_j \xrightarrow{\text{register}} BBA$ (somehow)

V_j gets $\boxed{f}, \boxed{g}, \boxed{h} \in_R G$ of size q
selects $\alpha_j \in_R \mathbb{Z}_q$, publish $h_j := \boxed{h}^{\alpha_j}$

(2) Pre-Voting Stage

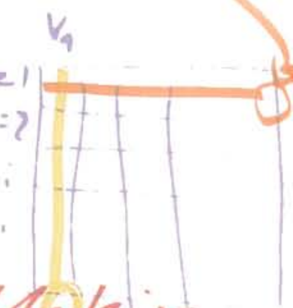
V_j: n values $s_{i,j} \in_R \mathbb{Z}_q$: $\boxed{\sum_{i=1}^{n} s_{i,j} = 0}$

$$\boxed{V_j \xrightarrow{R_{i,j},\ \boxed{R'_{i,j}}} BB}$$

$\boxed{R_{i,j}} := \boxed{g}^{s_{i,j}}$

$\boxed{R'_{i,j}} := h_j^{s_{i,j}}$

$\Rightarrow$ $n \times n$ Matrix



$\prod_i h_j^{s_{ij}} = h^{\alpha_j \underbrace{\sum_i s_{ij}}_{t_j}}$

$$BBA \xrightarrow{R_j := \prod_{i=1}^{n} \boxed{R'_{i,j}}} BB$$

(3) Ballot-Casting

$$\boxed{V_j \xrightarrow{B_j := h^{t_j} \boxed{f}^{\boxed{v_j}}} BB}$$

Vote: $\boxed{v_j}$

t_j unknown to the voter

(4) Tally

$$T := \prod_{j=1}^{n} B_j = \boxed{f}^{\sum_{j=1}^{n} \boxed{v_j}}$$

Kiayas & Yung   Hidden vote scheme!!

Eligibility : ✓ Check by signatures in the 1. stage

Verifiability : Individual : ✓ Everything open

global : ✓ Everything open

Anonymity : Yes, only one voter needs to supply random numbers!

Scalability : No! → $n \times n$ Matrix too big for many voters!

Receiptfreeness : No, possible to proof which vote was given

Robustness : No, but modification can solve this.

Questions : α_j's part ? ~~[illegible]~~

For obtaining h^{t_j}

A Verifiable Multi-Authority Secret Election Allowing Abstention from Voting.

By: Juang, Lei and Liaw.

P0

Initialization

Counter publishes all parameters, and signs them.

P1

Preparation

Administrators distribute secret shares to each other, and generate public keys and group public key.

P2

Global key generation

Scrutineers distribute secret shares to each other and generate public keys and group public key.

P3

Registration

Voters encrypt their votes using the group public key from P2, and apply unique blind signature technique to get their blind encrypted votes from P1.

P4

Voting

Voters generate their real encrypted votes from the blind encrypted votes received in P3 and send them to the counter via untraceable e-mail systems (Mix-net).

P5

Announcement

Counter publishes all accepted ballots.

P6

Publication

If no objections, counter requests arbitrary scrutineers to send him their shodow keys generated in P2. Then, counter computes the scrutineers' group secret key. Then, counter recovers the votes and publishes all real ballots.

Juang, Lei, and Liaw

eligibility - yes by using signatures

anonymity - ✓

verifiability - yes ✓

scalability - large scale

robust - mix-net

Coercion-Resistance Elec^2

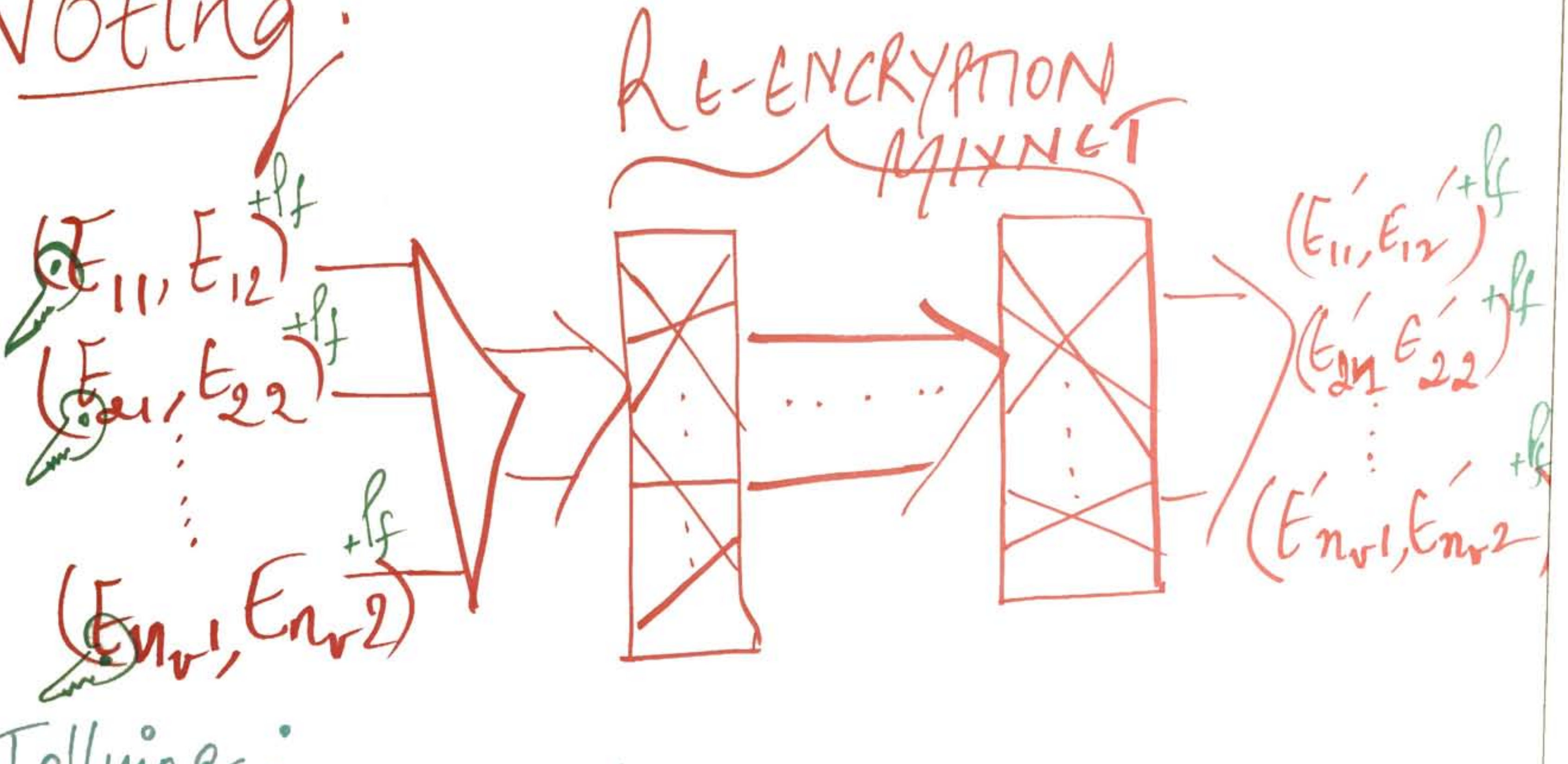
Juels-Catalano-Jakobsson

Registration:

$$(SK_R, \{i\}_{i=1}^{n_v}, k_1) \Rightarrow \boxed{\text{Registee function}} \Rightarrow \left(\{sk_i\}_{i=1}^{n_v}, \{pk_i\}_{i=1}^{n_v}\right)$$

Voter identifier

Voting:



Tallying:

$$(SK_T, BB, n_c, \{pk_i\}_{i=1}^{n_v}, k_2) \Rightarrow \boxed{\text{Tally Function}} \Rightarrow (X, P)$$

Verifying:

$$(PK_T, BB, n_c, X, P) \Rightarrow \boxed{\text{Verify func.}} \rightarrow \{0, 1\}$$

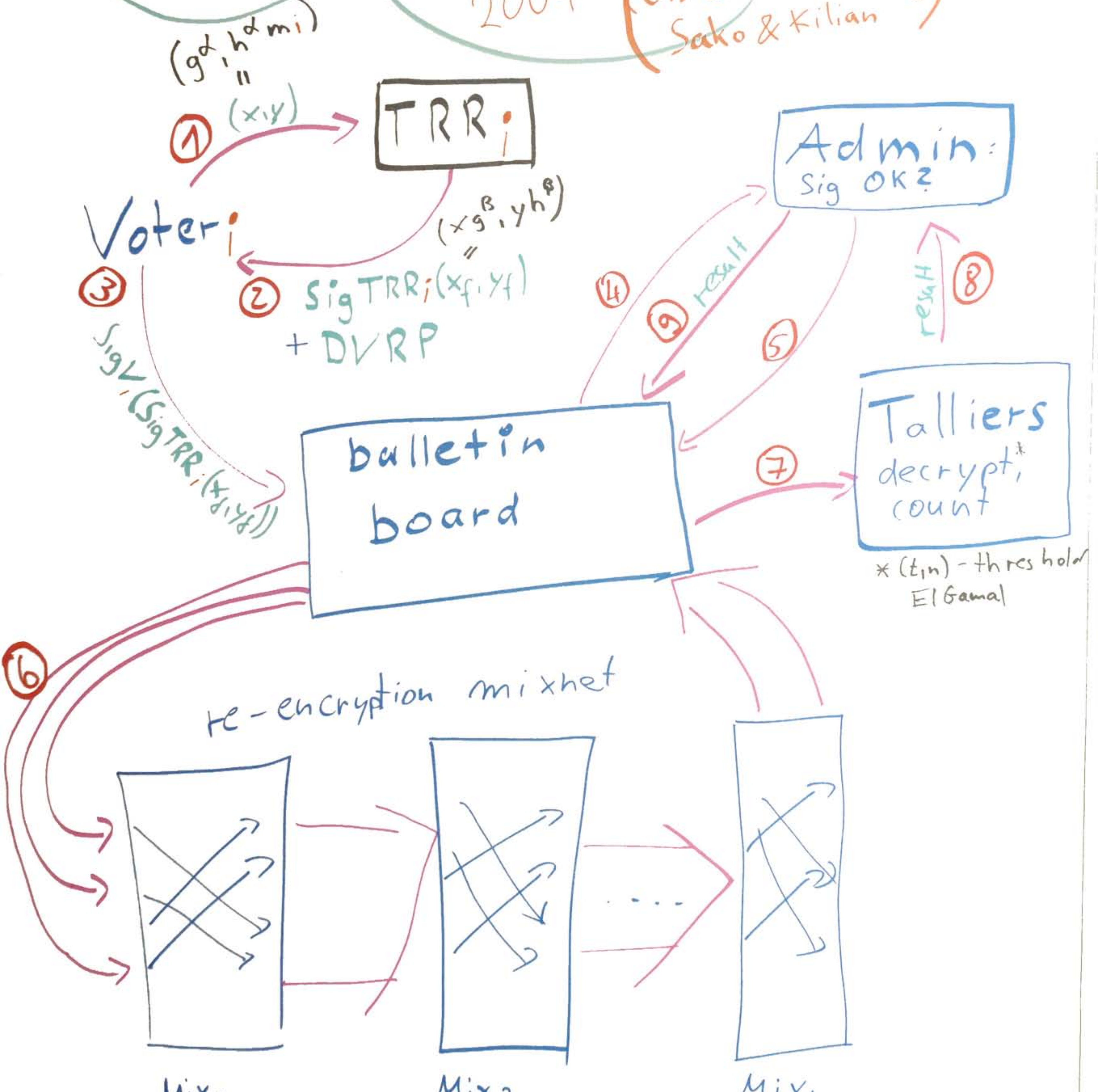
Voter V_i (on choice c_j and credential σ_i) for $a_1, a_2 \in Z_q$ sends through re-encryption mix net

$$E_1^{(i)} = (\alpha_1, \alpha_1', \beta_1) = (g_1^{a_1}, g_2^{a_1}, c_j h^{a_1}) \;\&\; E_2^{(i)} = (g_1^{a_2}, g_2^{a_2}, \sigma_i h^{a_2})$$

along with NIZK proofs (Pf).

Providing Receipt-Freeness in Mixnet-Based Voting Protocols

2004 (enhancemet of Sako & Kilian)

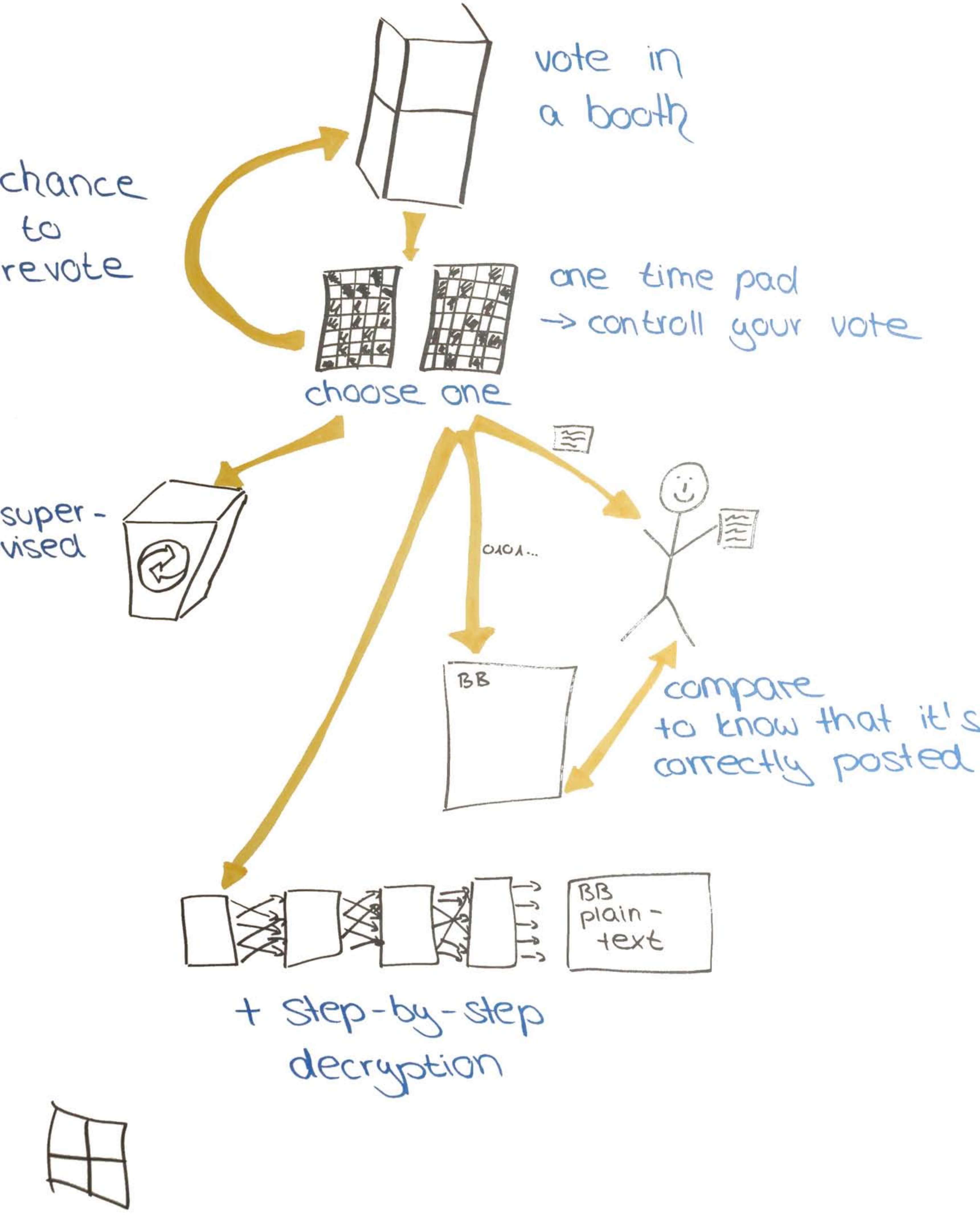


TRR: Tamper Resistant Randomizer | **DVRP**: designated verifier re-encryption proof

Lee, Boyd, Dawson, ... 2004
(enhancement of Karo & Kilian)

Questions: · No! :-)

Chaum 2004


vote in a booth
chance to revote
one time pad
-> controll your vote
choose one
super-vised
0101...
BB
compare to know that it's correctly posted
BB plain-text
+ step-by-step decryption

Chaum 2004

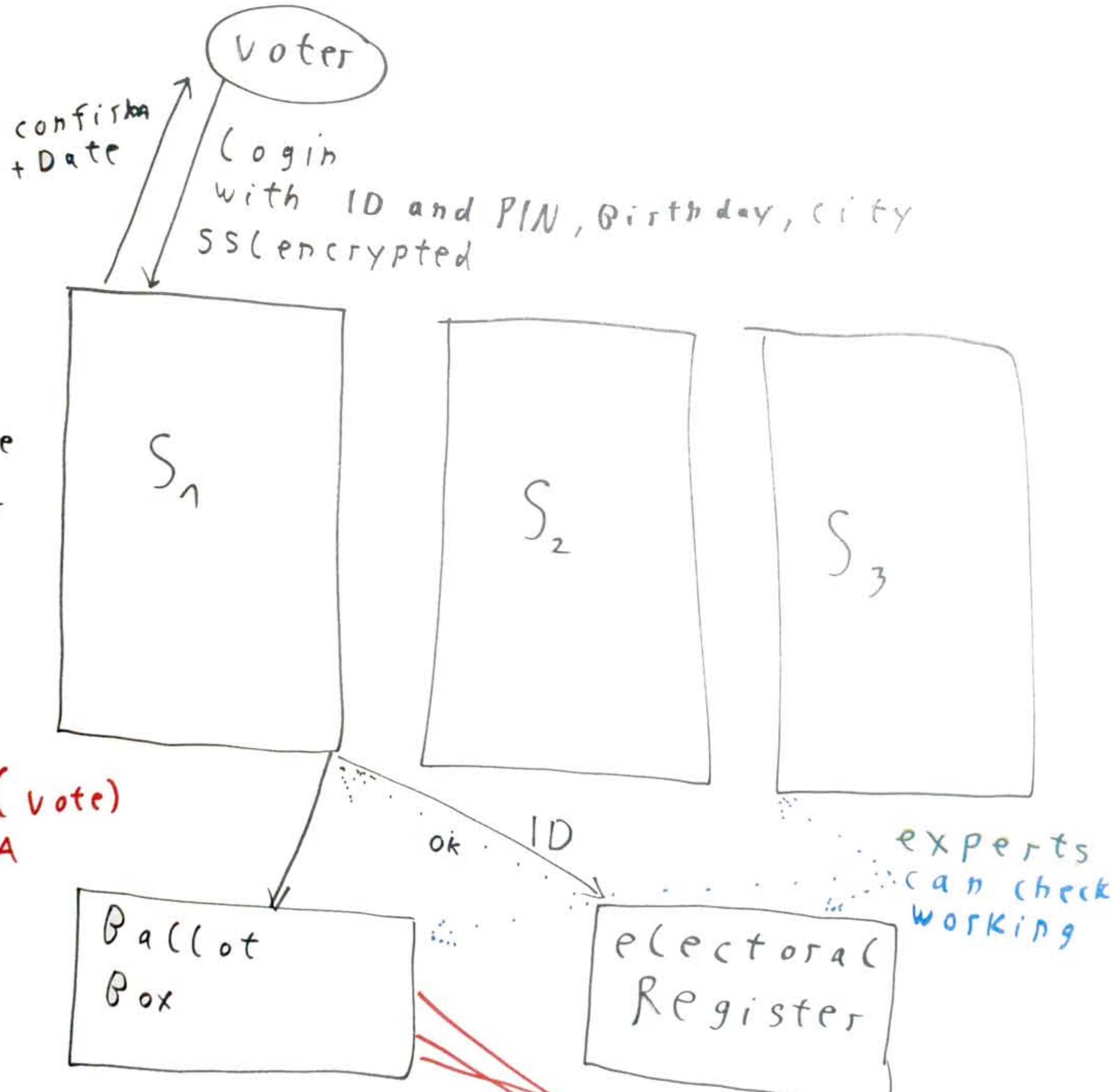
Questions. How it works?

Swiss (GENEVE)

Voting-Scheme

Registration: postal Letter with ID, PIN to every possible Voter

Voting:



Counting:

- Mix Ballot Box
- Decrypt all votes with $K_1||K_2$

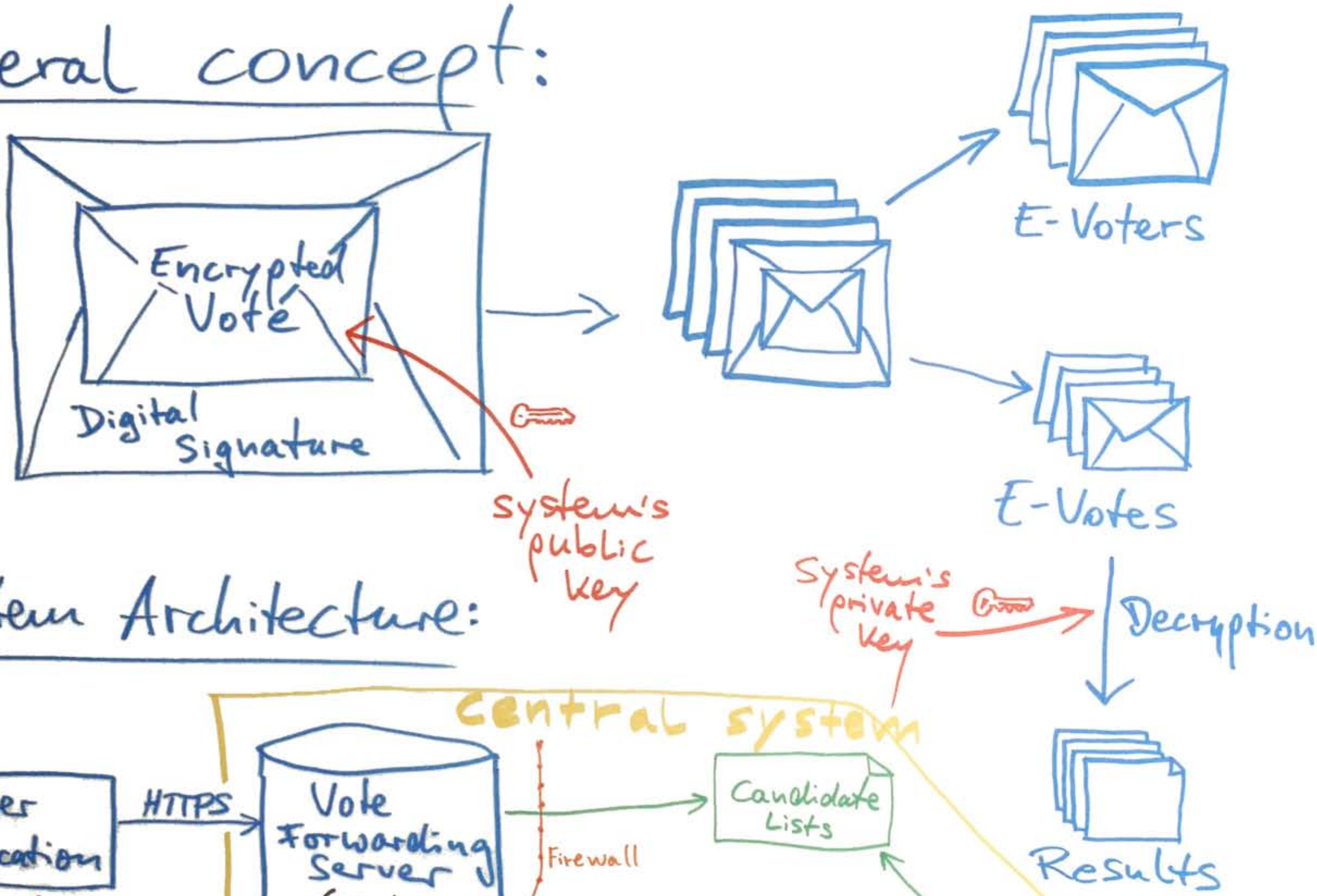
K_1, K_2 of different parties

$D_{K_1||K_2}$(votes)

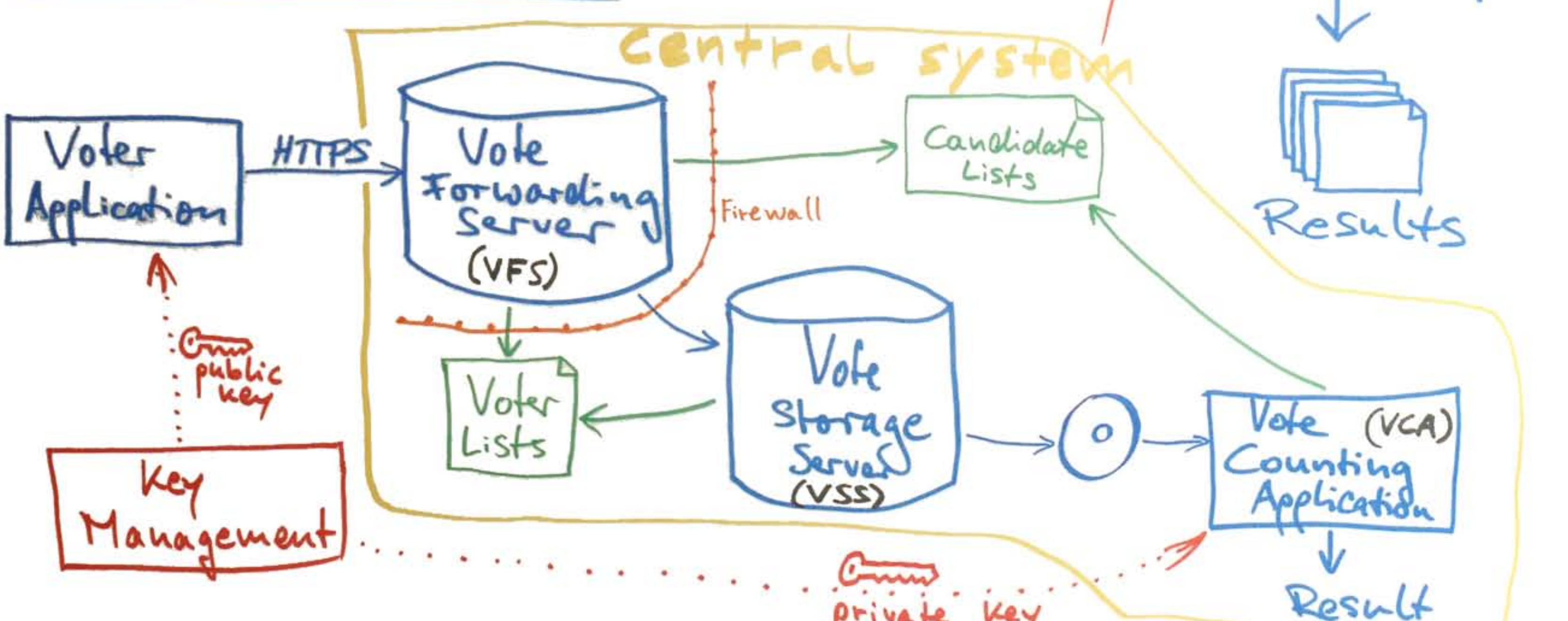
Results

Estonian E-Voting System

General concept:



System Architecture:



1) Voter: Encrypts and signs his/her vote and sends it to VFS
2) VFS: Authenticates Voter by his/her ID-Card and recieves vote
3) VSS: Recieves votes and seperates signatures from encrypted votes
4) VCA: Recieves only encrypted votes on CD (offline system) and performs counting of votes.

Considered systems, questions, left-overs

Chaum (1981). Simple systems, only covers basic desires. **Type**: Hidden voter.

Registration: Choose a random key pair. Send the decryption key anonymously (through a mixnet) to a bulletin board. The mixnet entry server only accepts one signed message per voter.

Voting: Encrypt the formatted vote with the encryption key. Send decryption key and the encrypted vote anonymously (through a mixnet) to a bulletion board. (The mixnet entry server can again control by requiring signatures that only voters send messages and only one. Yet, this can also be checked on the bulletin board.)

Tallying: Inspect the bulletin board! All votes are open.

- ◦ Eligibility: only eligible voters can vote and not more than once.
- ◦ Anonymity: as long as at least one mix is honest, the votes stay anonymous.
- ◦ Individual verifiability: Each voter can look for her decryption key on the bulletin board.
- ◦ Global verifiability: not provided, it is not clear that the mixes output the same things that they get.
- ◦ Receipts: a voter has a kind of receipt since only he knows the encryption key and can thus prove to a third person how he voted. His signatures which are available to the entry servers prove that she indeed sent the claimed messages.
- ◦ Robustness: a single mix blackout interrupts the entire system.

The system is a basis for many later constructions. The found problems can be resolved by additional measures, see followups.

Sako & Kilian (1995). Simple system, other key idea. **Type**: Hidden voter.

Registration: Each voter submits an encrypted ballot for each candidate through a re-encryption mixnet to a bulletin board. Each mix posts a proof of correct mixing and convinces the voter through an untappable channel how they permuted the ballots so that the voter knows which ballot on the bulletin board is for which candidate.

Voting: The voter submits the ballot for the desired candidate to a decryption mixnet. Each mix again posts a proof of correct mixing.

Tallying: Inspect the bulletin board! All votes are open.

- Eligibility: ok.
- Anonymity: as long as at least one mix is honest.
- Individual verifiability: Each voter can verify the proofs of correct mixing.
- Global verifiability: Yes.
- Receipts: There is a receipt of voting but no way to decrypt the encrypted vote.
- Robustness: a single mix blackout interrupts the entire system.

Cohen/Benaloh et al. (1985, 1986, 1987) Simple system, another idea. **Type**: Hidden vote.

Registration: Each voter submits an encrypted ballot for each candidate to a bulletin board, only the voter knows the order.

Voting: The voter submits the ballot for the desired candidate to a bulletin board.

Tallying: The votes are combined in encrypted form, the evaluation is then decrypted (by computing a discrete logarithm that it is known to be in a small interval).

- Eligibility: ok.
- Anonymity: yes (as long as used crypto is secure).
- Individual verifiability: yes.
- Global verifiability: yes.
- Receipts: yes.
- Robustness: yes.

Fujioka, Okamoto & Ohta (1993). More stages. **Type**: Hidden voter.

Registration: The voter commits (only) to his vote, this ballot is then signed blindly by an administrator who checks the eligibility.

Voting 1: The voter sends her ballot anonymously (through a mixnet) to the counter bulletin board.

Voting 2: The voter looks up her vote on the bulletin board and gets its serial number, she sends the commitment opening with the serial number again anonymously to the counter bulletin board.

Tallying: Inspect the bulletin board! All votes are open.

- Eligibility: ok.
- Anonymity: ok.
- Individual verifiability: ok.
- Global verifiability: ok.
- Receipts: The prover could possibly prove how she voted…
- Robustness: All entities could be distributed…Could they?

Kiayas & Yung (2002). Small elections, better security. **Type**: Hidden voter.

Registration: Each voter j selects a personal temporary key pair $(\alpha_j, h_j = h^{\alpha_j})$.

Pre voting: Each voter j selects a random number s_{ji} for all voters such that these add up to 0, and sends exponentiated values $(g^{s_{ji}}, h_i^{s_{ji}})$ to the bulletin matrix. The bulletin board multiplies the columns: $R_j := \prod_i h_j^{s_{ij}}$.

Voting: The voter j raises R_j to the α_j^{-1}-th power and multiplies this with f^{v_j}, the value $B_j = h^{\sum_i s_{ij}} f^{v_j}$ is posted on the bulletin board.

Tallying: All votes are multiplied, since the random numbers sum to 0 in each row and thus in total, the exponents of h combine to 0, we are left with $f^{\sum v_j}$. Since we know that the exponent is small, this discrete logarithm can be computed.

- Eligibility: ok.

- Anonymity: ok, unless all other voters coalesce.
- Individual verifiability: ok.
- Global verifiability: ok.
- Receipts: None. (?)
- Robustness: The scheme can be modified to tolerate absent or abstaining voters.

Juang, Lei & Liaw (2002). **Type**: Hidden voter.

Registration: The voter encrypts her vote and gets a blind signature from an administrator.

Voting: The voter sends her encrypted vote anonymously (via a mix net) to a counter bb.

Tallying: The counter publishes the encrypted votes. If there are no objections, the scrutineers jointly decrypt the votes and the open votes are published on a bulletin board.

- Eligibility: ok.
- Anonymity: ok.
- Individual verifiability: ??
- Global verifiability: ??
- Receipts: ??
- Robustness: ??

Juels, Catalano & Jakobsson (2005). **Type**: Hidden voter (and hidden vote?).

Registration: Each voter gets a temporary key pair certified.

Voting: Each voter encrypts her vote and sends it anonymously via a re-encryption mixnet to a bulletin boards. The voter proofs that she correctly encrypted and the mixes that they correctly mixed and re-encrypted.

Tallying: All votes are combined, the tally and a proof of correct tallying are posted.

Verification: Anybody can use the publicly available information to check the global correctness.

- Eligibility: ok.
- Anonymity: ok.
- Individual verifiability: ??
- Global verifiability: ok.
- Receipts: ??
- Robustness: ??

Lee, Boyd, Dawson, Kim, Yang & Yoo (2004). Type: Hidden voter.

Registration: Each voter registers and obtains a tamper resistant randomizer, say a smart card.

Voting: The voter encrypts her vote, re-encrypts and signs it using the tamper resistant randomizer. The device also provides a proof of correct reencryption. The re-encrypted vote and the device' signature are posted to a bulletin board. Its admin checks the signature.

Tallying: A re-encryption mixnet anonymizes the content of the bulletin board and proves correct mixing. The talliers (a decryption mixnet) decrypt and count.

- Eligibility: ok.
- Anonymity: ok.
- Individual verifiability: ok.
- Global verifiability: ok.
- Receipts: None.
- Robustness: Can be added by using robust mixnets.

Chaum (2004). Not entirely electronic. **Type**: Hidden voter.

Voting: The voter has a device encrypt the vote, chooses a few bits during this encryption. A device does that and prints two slides that overlaid as a visual cryptogram show the vote. Finally, the voter chooses one half of the visual encryption to be passed on. The device signs that half. The other half is desctroyed under supervision. In particular, the device cannot manipulate the printout when it has to sign. The signed ballot is posted on a bulletin board that can be checked by the voter using his share.

Tallying: All votes are decrypted by a mixnet and posted on a bulletin board.

We are missing quite a few details, maybe checking Jakobsson, Juels & Rivest (2002) would reveal the concept.

- Remote: NO.
- Eligibility: ok.
- Anonymity: ok.
- Individual verifiability: ok.
- Global verifiability: ok.
- Receipts: ok.
- Robustness: Implementable.