<table>
<thead>
<tr>
<th>Time</th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>4</td>
<td>3</td>
<td>Cross</td>
<td>Cross</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>Cross</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>Cross</td>
</tr>
<tr>
<td>14</td>
<td>Tutorial</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>16</td>
<td>Tutorial</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>18</td>
<td>1/2</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

Elective?

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

One solution: a dictatorship.

Make a decision!
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
- geheim - secret
- allgemein - universal (no restriction by race, gender, belief, social status)

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

Non-issue: Not 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
Survey of election voting technology

- Australian paper ballot
- 750 Ancient pieces of broken pottery
- 138 Romans 'gabinia lex'
- (paper) ballots for elections of magistrates
- 1629 17A 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.)

1549 Ballot from...palle ... Ballot of palla "ball". Earliest reference due to Venice.

1838 Britain Chartist petition among other things asks for secret ballots.

1854 Australia Influenced by Chartists, revolving mirrors in Victoria's adopt secret ballot (as part of the entire Charter).

1856 Australia Shakes Tasmania, Victoria and South Australia each legislation.

New South Wales (1858), Queensland (1859)
Western Australia (1877) followed.

1870 New Zealand
1872 Britain  'Ballot Act' introduces secret ballots. This reduces substantially the cost of campaigning.

1874 Canada
1884-85 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

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.

(Shortviewed 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.

Lever voting machines

1892: NY, first use
1930: all larger urban centers

Pros / 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

1864: 1877's Postal punch card mechanism, Ars T in Georgia.
1972: 100%
1988: roughly 1/3

Pros: No bias in counting. (?)
Pro: Test election result.
Pro: Backup available.

Con: Punch often not clean. (1.5%)
Con: Repunched chads or badly punched ones may fall, especially during recount.
Con: No intuitive way to decide whether a vote is valid.
**Optical Mark-Sense Voting**

1970s ... 
1988: about 1/4 of US voters used this.

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

**Accuracy:** Manual barcodes: $10^{-5}$ (for perfectly marked sheets)

2000 Florida election: 1 in 2000 votes were problematic.

$\Rightarrow$ 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

Central count

type

decentralized

centralized

supervision

difficult

easy

counting observers

necessarily allowed at each precinct

easy

transmission

accuracy?

security? [ballot box theft, ...]
faster?

overvotes

not a problem

hand examination

Voting paradox

Arrow's theorem/

There is no "public welfare function"

\[
\text{set of voters} \rightarrow \text{Ranking}
\]
such that the following hold:

- no dictator
- some - each alternative is relevant
- is a hill climber
- no tactical voting
Remotes voting

→ matter of law

→ many enabling technologies used so far.

vot
early voting

On election day
you can vote
wherever you like.
(Not a problem
because of high
susceptibility to fraud.)

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

Con: often many different ballot styles

Johnson County, Iowa, 100,000 inhabitants
→ 70 distinct ballot style!

Con: fraud is easier

→ family voting
→ selling blank absentee ballots

Best known defense: Allow revote.
Many different criteria!

- Legal criteria
- Voter acceptance? Usability?
- Whom do we have to trust?
  - How is the system constructed?
  - ... administered?
  - Open source?
  - Oversight and audit trails?
  - Monitoring (by opposing parties)?
  - Cryptography? Electronic signatures?
  - Modularity? Redundancy?

- Compatibility?
  - Integration into larger systems?
  - Open standards (for ballot forms, ... ) vs. complete replacement

Many more details once you start to work on details.
Cryptography
primitives

Tool: modular arithmetic,
ring of integer modulo \( N \) \( (N \in \mathbb{Z}_2) \)

\[ \mathbb{Z}_N = (\mathbb{Z}_N, +, \cdot) \]
\[ (\mathbb{Z}, +, \cdot) \]
\[ \{ 0 \pm 1 \} \]

\( \mathbb{Z} \) has division with remainder.
Given \( a, b \in \mathbb{Z} \) with \( b \neq 0 \)
and \( q, r \in \mathbb{Z} \) 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:

\[ 3y \pmod{2} : x, y = 1 \pmod{2} \]

Do we a question in \( \mathbb{Z} \)?

\[ 3y, q : x \pmod{2} \cdot y + (-q) \cdot y \pmod{2} = 1 \]

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

\[ s \cdot a + t \cdot b \] is small!

\( \text{and positive} \)

\( a = 35, b = 31 \)

<table>
<thead>
<tr>
<th>( i )</th>
<th>( r_i )</th>
<th>( q_i )</th>
<th>( s_i )</th>
<th>( t_i )</th>
<th>( \text{Comment} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| 1 | 11 | 3 | 0 | 1 | \( 35 = 1 \cdot 35 + 0 \cdot 11 \)
| 2 | 2 | 5 | 1 | -3 | \( 11 = 0 \cdot 35 + 1 \cdot 11 \)
| 3 | 2 | 2 | -5 | 16 | \( 2 = 1 \cdot 35 - 2 \cdot 11 \)
|   | 0 | 11 | -35 | 0 | \( 1 = -5 \cdot 35 + 2 \cdot 11 \)

In particular, \( 16 \cdot 11 = 1 \pmod{35} \).
Thus, 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 \( O(n^2 \log q) \)).

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 \( \log q > 0 \)) there does not exist a solution of either equation.
First primitive:
public key encryption scheme

Example: RSA (Rivest, Shamir & Adleman 1978)

Situation

\[ y = \text{enc}_{\text{public}}(\text{msg}) \rightarrow z = \text{dec}_{\text{private}}(y) \]

Hopefully: \( z = \text{K} \)

Correctness

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

Efficiency

\( \text{a) polynomial time} \)
\( \text{(in the asymptotic setting)} \)
\( \text{b) within 'seconds'} \)
\( \text{(in the real world)} \)

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

**Input:** a security parameter n

**Output:** a public key $\langle N, e \rangle$ and a private key $\langle N, d \rangle$

1. Generate a $\frac{n}{2}$-bit prime $p$.
2. Generate a $\frac{n}{2}$-bit prime $q$.
3. $N \leftarrow p \cdot q$.
4. $L \leftarrow (p-1) \cdot (q-1)$ “repetition length” $L = \varphi(N)$
5. Choose $e, d$, $0 < e, d < L$, such that $e \cdot d = 1$ in $\mathbb{Z}_L$.
   (Use $\varphi$ EEA!)
6. Return $\langle N, e \rangle$, $\langle N, d \rangle$
   - public key for encryption
   - private key for decryption

**RSA encryption**

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

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

1. $y \leftarrow x^e \mod N$.
2. Return $y$.

**RSA decryption**

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

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

1. $z \leftarrow y^d \mod N$.
2. Return $z$. 
Correctness?

\[ z = y^d \]
\[ y = x^e \quad \text{in } \mathbb{Z}_N \]
\[ z = x^{ed} \]
\[ = x^{1 + k \cdot L} \]
\[ = x \]

because we have \( x^L = 1 \)

in case \( x \) is invertible in \( \mathbb{Z}_N \)

by the Theorem of Lagrange, or Euler.

Thus (Lagrange)

Given a group \((G, \cdot)\), finite, (commutative),

we have \( x \# G = 1 \quad \text{in } G \)

for any \( x \in G \).

We use \( G = \mathbb{Z}_N^x \) unit group of

the ring \( \mathbb{Z}_N \) of

in least modules \( N \).

Its elements are all invertible elements of \( \mathbb{Z}_N \).

Its operation is the multiplication inherited from \( \mathbb{Z}_N \).

\[ \text{P. } \text{Given } a, b \text{ invertible in } \mathbb{Z}_N \text{ check that } a \cdot b \text{ is invertible.} \]

\[ \text{N. } \text{check that } x \in \mathbb{Z}_N^x \]

\[ \text{1. } \text{given a invertible, take } b \% \text{ abs } 1, \text{ check that } b \text{ is invertible.} \]

\[ \text{C. } \text{unit } x \text{ is an invertible element (in a ring).} \]
Theorem of Euler

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

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

Ad hoc:

Which elements are \( \text{not} \) invertible?

\[
0, \ p, \ 2p, \ 3p, \ldots, \ (q-1)p, \ q, \ 2q, \ 3q, \ldots, \ (p-1)q
\]

These are all different!

If \( ap = \beta q \), then because \( p, q \) are different primes:
\( q \mid a \) and \( p \mid \beta \), so \( ap = \beta q = 0 \).
Thus \( a = 0 \) and \( \beta = 0 \).

And no other is non-invertible!

Given \( (x, p, q) \in \mathbb{Z}^3, \ p, q, p\cdot q \), \( 0 \leq x < N \).

If \( q > 1 \) then \( x = \alpha p \Rightarrow \)
\( \alpha \in \mathbb{Z}_q = \{0, 1, \ldots, q-1\} \) e.g. \( x = \beta q \),
\( x = \gamma \) e.g. \( x = 0 \).

Thus \( \# \mathbb{Z}_N^\times = N - 1 - (q-1) - (p-1) = \varphi(N) = \varphi(\varphi(N)). \)
Thus we have
\[ z = x \]
whenever \( x \) is invertible in \( Z_N \),
\[ \text{i.e. } \gcd(x, N) = 1. \]
Actually, also if \( x \) is not invertible
one can prove \( z = x \).

But then 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 \( 2^{-\sigma} \).
That's practically zero.

Efficiency?
\[
\text{enc/dec: square & multiply over } Z_N
\text{ (repeated squaring) } \rightarrow O(n^3)
\text{(this is poly-time and about a second for } n = 512 \text{ on a real computer)}
\]
\[
\text{key generation: } O(n^4) \text{ for generating primes}
\text{everything else is cheaper...}
\]
Security?

Complete breaks:
1. Factor \( N \) if possible.
2. Find \( L \).
3. Find \( d \).
4. Find \( x \).

Actually:

\[
\frac{(t-p)(t-q)}{T^2 - (N-1+t)T + N}
\]

Suppose you solve (3) twice:
- \( e_1 d_1 = 1 \), \( e_2 d_2 = 1 \)

\[
\text{gcd} (e_1 d_1 - 1, e_2 d_2 - 1)
\]

with high probability

\[
\text{gcd} = 1 \text{ or } 2 \cdot 2
\]

Open question: (4) \( \Rightarrow \) (3)?

Still, this is much less than wanted!

Another break:
(5) Find \( b \cdot \phi(x) \) given \( (N, e, y) \).

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

"(5) \( \Rightarrow \) (4)".

Case \( \ell' \cdot \sigma(x) = 0 \):
\[
\begin{align*}
\ell' &= 2x \cdot e^x \\
y &= x^e - 1
\end{align*}
\]
If we had $(5) \implies (7)$
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 be:

1. Identify signer
2. "Identify" the document (no changes)
3. Obtain signer and document

ElGamal signatures (and DSA or ECDSA)

Key generation:

Choose a large prime $p$ (e.g., 2048 bit)
and a large prime $q$ (160 bit)
such that $q | p - 1$.

Choose $g \in \mathbb{Z}_p^*$ such that
$g^q = 1$ and $g \neq 1$.

(To do so choose $k \in \mathbb{Z}_p^*$ arbitrarily
and compute $g = h^{\frac{1}{k}}$ until $g \neq 1$.)

Fix a hash function $h : \{0, 1\}^* \rightarrow \mathbb{Z}_p^*$.

Choose a private key $x \in \mathbb{Z}_{p-1}^*$.

Compute the public key $a = g^x \mod p$. 
Signature verification:

Input: a message, \( b = (b_1, b_2) \) signature, a public key of signer.

Output: Accept (1) or Reject (0)

1. Verify \( b \in \mathbb{Z}_p^\times \), \( g \in \mathbb{Z}_p \) and \( b \cdot g \equiv \text{hash}(m) \mod q \).

2. Check \( a \cdot b = b^* \)

where \( *: \mathbb{Z}_p^\times \to \mathbb{Z}_q \) has almost no structure and is very easy to compute (and is almost surjective).

For example, if \( a \cdot b \in \mathbb{Z}_q \), then \( a \cdot b \mod q = b \cdot g \).

How could we solve the equation for \((b_1, b_2)\)?

First try: choose \( b \in \mathbb{Z}_p^\times \) and solve for \( g \):

\[ b \cdot g \equiv \text{hash}(m) \mod q \]

This is a so-called discrete logarithm problem \((1, 3, 5, \ldots)\) in our case in \( \mathbb{Z}_p^\times \), or in \( \langle g \rangle \).
Second try: Choose \( y \in \mathbb{Z}_p \) and solve the key equation for \( b \):

\[
\begin{align*}
\alpha^b &\equiv y \pmod{p} \\
\alpha &\equiv \text{hash}(m)
\end{align*}
\]

That looks even weirder.

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

\( \rightarrow \) brute force! \( \rightarrow \) time \( O(p^2) \)

\( \rightarrow \) birthday? if possible: clever

\( \rightarrow \) time \( O(V_q) \)

at least

\( \text{still } V_q \approx 2^{200} \)

Third try: Try to solve for \( b, \alpha \) simultaneously ...

Even less ideas ...

How to solve with extra knowledge?

Note that \( \alpha = \text{hash}(m) \) known to signer!

Look at the key equation

\[
\begin{align*}
\alpha^b &\equiv g \pmod{p} \\
\alpha &\equiv \text{hash}(m) \\
\text{power of } g &? \quad \text{Make it a power of } g!
\end{align*}
\]
signature generation:

Input:  
- global setup, 
- a private key, 
- a message.

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

1. Choose \( \beta \in \mathbb{Z}_q \) and compute \( b = g^\beta \) in \( \mathbb{Z}_p^* \).

2. Solve the key equation:

\[
\alpha b^x + \beta \delta^x = \text{hash}(m) \quad \text{in} \quad \mathbb{Z}_q^*
\]

for \( \delta^* \), or rather equivalently

\[
\alpha b^x + \beta \delta^x = \text{hash}(m) \quad \text{in} \quad \mathbb{Z}_q
\]

i.e. \( \delta = \beta^{-1} (\text{hash}(m) - \alpha b^x) \in \mathbb{Z}_q \) by EEA

3. Output \((b, \delta)\).
A little background

exponential group
(for \( \mathbb{Z}_p^x = \mathbb{Z}_p \))

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

Thus \( g = h^{q-1} \) 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 \( 1 \neq g = g^{k \cdot s + q \cdot t} = (g^k)^s \cdot (g^q)^t = (g^k)^3 = 1 \)

if we had \( g^k = 1 \). So \( g^k \neq 1 \).
When we consider the map

\[ \exp_g : \mathbb{Z}_q \rightarrow \mathbb{Z}_p^x \]

\[ \alpha \rightarrow g^\alpha \]

(This is: \( \mathbb{Z}_{p-1} \rightarrow \mathbb{Z}_p^x \)

\[ \beta \rightarrow h^\beta \]

after first multiplying with \( p^{-1} \) : \( \beta = \rho a \cdot p^{-1} \).)

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

effective.

The image are precisely the powers of \( g \),
which form a subgroup \(<g>\) of \( \mathbb{Z}_p^x \).

\[ \{ 1, g, g^2, \ldots, g^{q-2} \} \]

Computing values of the map \( \exp_g \) is easy:

square & multiply! \rightarrow O(n^3)

But: the inverse map, the discrete logarithm

\[ \mathbb{Z}_q \leftarrow <g> \]

\[ \alpha \leftarrow g^\alpha \]

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 assumptions are:

- the DLP for \( \mathbb{Z}_p^x \) with \( p \) prime is difficult (theory: no poly-time, practice: \( p \) at least 1024 bit)
- the DLP in an elliptic curve group \( E \) 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 banned.

Does the ElGamal scheme fulfill our requirements?
Correct? Yes, by construction: a signature solves

\[ a^{b^x} \cdot b^y = g \]

which is equivalent (!) to

\[ \alpha b^x + \beta \delta = \text{hash}(m) \]

The \((b, g)\) is a signature to \(m\).

Efficiency? Yes, only prime generation, group operation, modular computations, \((\text{FST})\), square & multiply.

Security?
\[ \text{What is the task of an attacker?} \]
gets: \(\alpha\), global setup: \(\mathbb{Z}_p, g, \text{hash}, \ast, \ldots, \)
\[ q = \#<g> \]

Additionally, the attacker may read or even choose messages that will be signed by \(\alpha\) or others.

Task: Output a message with a valid signature, which was never queried.
For example, what does non-existence of such a poly-time restricted attacker imply for the hash function?

\[
\text{hash: } \{0,1\}^* \rightarrow \mathbb{Z}_q \text{ in our scheme.}
\]

If there exists a poly-time algorithm that outputs \( m_1, m_2 \in \{0,1\}^* \) not such that \( m_1 \neq m_2 \),

\[
\text{hash}(m_1) = \text{hash}(m_2)
\]

(Not: \((m_1, m_2)\) always exist, but are possibly difficult to find.)

An attacker could use such a pair to ask for a signature of \( m_2 \) and return \( m_1 \) with this signature. This contradicts security requirement.

**Conclusion:**

\[
\text{hash not collision-resistant } \Rightarrow \text{ ElGamal with hash is broken.}
\]

\[
\text{ElGamal using hash is secure } \Rightarrow \text{ hash is collision-resistant.}
\]
(1) Identify signer? (not knowing \( \alpha \))

Something else cannot produce a valid signature because he would be an attacker in the sense of our security goal. So a 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 continued

1. Secure channels
   - Combine encryption and signatures.

2. Untappable channels
   - Physical security.

3. Anonymous channel
   - Nikets
     - Many messages
     - Several
     - Demuelenbeck
     - Declining cryptographers

   - DC-net (for the intended recipient)
Decryption mixnet

Sender i:

Input: $m_i$, global setup.
Output: $c_i$.

1. Choose random string $r_i$.
2. Encrypt $m_i$ with $r_i$ with the public key of $Mix_2, Mix_3, \ldots, Mix_4$:
   
   $$c_i = E_{1} \left( E_{2} \left( \ldots E_{e} \left( m_i ; 11r_i \right) \ldots \right) \right)$$

3. Return $c_i = \pi c_i^{10}$

$Mix_3$:

Input: a list of messages $\left\{ c_{i}^{(i-1)} \right\}$.
Output: a list of messages $\left\{ c_{i}^{(i)} \right\}$.

1. Decrypt all messages with i's private key:
   
   $$s_{i}^{(i)} = D_{i} \left( c_{i}^{(i-1)} \right)$$

2. Sort the new list:
   
   $$(c_{i}^{(i)}) = \text{sort} \left( s_{i}^{(i)} \right)$$

3. Return $\left( c_{i}^{(i)} \right)$.

Receiver: sets $(m; \pi)$. 
- No relation between output message position and sent message position.
  - Randomness important!
  Otherwise re-encrypting reveals the sender.
So the messages do not reveal the sender and thus we have anonymity.

Problems:
  - Can a corrupt Prix or several recover the connection message/sender?

  NO, if encryption is INDISTINGUISHABLE Security goal.

Security goal
There is no poly-time attacker that can solve the following game:

Input: setup.
Intermediate output: msg₁, msg₂ of same length.
Intermediate input: Encryption of msg₁ with i ∈ {0, 1}.
Output: a bit j.
  - Additional query a decryption oracle.
The wins if i = j and he never passes the challenge.
RSA itself does not fulfill that:

It could compute \( E(m_{1}) \) and \( E(m_{2}) \) 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 factors or prime itself.

**Encrypt** input: \( m \in G \), message, \( a \) = public key of recipient.

Output:

1. Choose \( k \in \mathbb{Z}_{q} \).
2. Return \( (g^k, a^k \cdot m) \).

**Decrypt** input: \((w, y) = (g^k, a^k \cdot m), x \) = private key.

Output: \( m \)

Return \( y / w^x = a^x/g^x = q^x \cdot x \mod q \).
So if at least one Mix remains uncorrupted, the entire process stays anonymous.

Danger: the mix net does not provide confidentiality or integrity.

Clear: it cannot give authenticity.

Re-encryption mix net

**Mixj**; Input: list of cipher texts $(g^{s_i}, a^{s_i} x_i)$

Output: $- - -$ (solkh)

1. Re-encrypt:
   - choose $x_j = y_j - s_{ij-1} \cdot r \in \mathbb{Z}_q$
   - calculate $g^{\delta_j} = g^{x_j} \cdot a^{x_j} \cdot g^{s_j}$
   - $g^{x_j} \cdot a^{x_j} \cdot g^{s_j} = (g^{s_j}, a^{s_j}, g^{x_j-1}, a^{x_j-1})$

2. Sort.

3. Return list.
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

+ Some anonymity features even better - because of new randomness in each stage.
Secret world

\[
\begin{align*}
x = g^x \\
\text{Baby Step Giant Step} \\
g \cdot x = g^0 \rightarrow O(Vq) \\
\text{Pollard's H"ellman} \quad q = p^x + 1 \\
x^x = (g^x)^x \\
\varepsilon = \varepsilon_0 + \varepsilon_1 p_1 + \varepsilon_2 p_2^2 \\
\rightarrow \text{Decompress } \varepsilon_0
\end{align*}
\]
ElGamal signatures

Verify:

\[ b^* \quad b^y = \text{hash}(m) \quad g^r \quad \text{in } G \]

Signed?

\[ a = g^x \]

\[ m_x(b^y) : b^* = (g^{\text{hash}(m)})^x \]

Sign?

\[ B = g^B \]

\[ a^x b^* + B^y = \text{hash}(m) \]

Solve and answer: \((b, y)\).

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

User Setup:
Choose private key \(a\).
Compute public key \(a = g^x\).
Elgamal encryption

Sender
\[ m \in \mathbb{Z}_p^x \]
choose \( s \in \mathbb{Z}_q \).

\[
\begin{align*}
(g^s, a^{sx}) &\rightarrow (r, y) \\
\end{align*}
\]

Recipient
\[ a = g^\alpha \]
\[ z = \frac{y}{r^\alpha} \]

\[
\begin{align*}
z &= y \cdot y^{\alpha} \\
&= g^{sx} \cdot g^{sx} \\
&= x \quad \checkmark
\end{align*}
\]
Given encryption scheme \( g, r \)

\[
\text{setup} \leftarrow g, r
\]

\[
\text{enc} \alpha
\]

\[
\text{dec} \alpha
\]

\[\text{IND-CCA}\]

- Indistinguishability
- Chosen Ciphertext Attack

Public setup \( g, r \)

\[
\text{Dec}
\]

\[
\text{Enc}_i\text{ed} + x, z
\]

\[
\text{Enc}(\text{msg};)
\]

\[
\text{msg}_1, \text{msg}_2
\]

\[
\text{j}
\]

\[\text{Win} \neq i = j ?\]

**RSA does not have this!**
Indistinguishability
key only attack

Reduction

\( (g, g^x, g^y)^r \mod g \)

\[ y = \alpha \beta \]
Classification (rough):

- Hidden voter
- "anonymous submission of vote"
- Hidden vote
- "encrypted submission of vote"
- Hidden voter with hidden vote

Hidden voter

Scheme in Chaum (1989)

Announcement stage

- Chaum's decryption mixnet and its RSA public parameters: \( E_k (m, r) = E_{k_1} (E_{k_2} (\ldots E_k (m, r) \ldots)) \)
- Each vote is associated with a digital signature.

Registration stage

1. Token generation: The eligible voter generates a random RSA key pair \( k_j \) (public key) and \( \bar{k}_j \) (private key) and sets \( \text{token}_j = k_j \).
0. Voter \( V_j \) sends the token \( j \) in encrypted form to Mix, as

\[
E_k(\text{token}_j, V_j)
\]

and a digital signature on \( \text{token}_j \) to prove eligibility.

Mix sends a receipt to \( V_j \) and process through the mixnet.

1. Mix outputs a lexicographically ordered list of voters tokens (token\(_j\)) to a bulletin board.

**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

\[
E_k(\text{token}_j || E_{k_{\text{mix}}}(V_j \| 0^k), V_j)
\]

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

After mixing, Mix outputs a lexicographically ordered list of \( V_j \| E_{k_{\text{mix}}}(V_j \| 0^k) \) on the bulletin board.
Eligibility

Only eligible voters can vote and cast votes.

This is granted if the voter is not corrupted.

To guarantee that the system works correctly, we could add that it has to prove that it only sent a single per voter eligible vote and no token for non-eligible ones. Maybe a further bulletin board could do that.

This makes control of

(*) one eligible voter ⇔ 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, and thus make that token invalid and later register a new one.

This way, in the voting stage again the inputs to Mix will 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.

(*) $E_{UN}(m, r) = E_{K_1}(E_{K_2}(\ldots E_{K_0}(m \parallel r) \ldots \parallel r_2) \parallel r_1)$

so that no mix can reconstruct $r$. 
Anonymity

Granted if at least one ticket remains uncorrupted.

Verifiability

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

General: ✓

Receipt-freeess

NO!
- Not fair, vote-selling and family voting are possible.
Announcement stage

Set up for a re-encryption mixnet:

- a group $G$ (e.g., $G = \mathbb{Z}_p^*$, $\#G = q$, $p$ prime, $g \in G \setminus \{1\}$. \[ \Rightarrow \ q | (p-1) = \#\mathbb{Z}_p^* \])

- a key pair $(S_i, s_i)$ in $K$: $S_i = g^{s_i}$

for Mixi, $1 \leq i \leq e$

and $K = \prod S_i = g^{\sum 6}$.

Registration stage

The eligible voter $V_j$ registers and interacts with the mixnet:

Input: $E_k(v(f), 0) = (g^v, K^v f^v(f))$

for $f \in$ SetOfCandidates.

1. Mixij chooses a permutation $\pi_{ij}$ and commits to it to voter $V_j$.
   (i.e., give locked boxes with $\pi_{ij}$ in them to voter $V_j$).

2. Re-encrypt with a random shift $\pi_{ij}$

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

   $s = s' + \pi_{ij}$

   - For each voter $V_j$
   - For each mix $\text{Mix}_i$
3. Permute these re-encrypted votes using \( \pi_{ij} \) (varying the \( f \)'s).

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

5. Decommitt \( \pi_{ij} \) to votes \( V_j \) over an unopeable channel, which verifies \( \pi_{ij} \) using the posted proof.

Output: \( \{ E_K(v_i, \pi_{ij}) \mid i \in SelOSCandidates \} \)
1. **Voting stage**

\[ \nu_j \text{ chooses one of the 50 channels from the output of the mixnet, and sends it.} \]

2. **Mixing**

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

\[ E_k(\nu^{(c)}, r_{j,f}) \]

\[ (g^s, \nu^{(c)}) \]

\[ (g^{s_1}, s_2^{s_3} \cdots s_n^{s_j}) \]

\[ V_i = \Pi s_i; \quad \Sigma c_i; \quad = g \]

BB with proofs

**non-intrusive proofs of correct mixing**
Eligibility

As in Charm (1987),

one voter $\rightarrow$ one vote!

Anonymity (privacy)

Given, relies on the unappable channel.

But then anonymity is even unconditional.

If you replace the unappable channel with a private channel (so using encryption and signatures) then anonymity is only computational.

Verifiability

Individual:

$\checkmark$ correct tokens?

all votes counted?: $\checkmark$ Yes!

own

(see if the proofs of correct mixing in
the mixing in the voting stage)

General:

$\checkmark$

Fairness

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

Accuracy

Everything works correctly and provably so.

Verifiability: No one can compute a partial tally.

Robustness

Decryption mix must be known by
Hidden vote

\[ \text{Volt}^* \]
\[ \text{proof} \]
\[ \text{Authority:} \]
\[ \text{first "multiplies"} \]
\[ \text{the encrypted votes and puts} \]
\[ \text{EN} \sum v_j \]
\[ \text{then decrypt this!} \]
\[ \text{Output \sum v_j} \]
\[ \text{and proofs...} \]

Benaloh & Fischer '85: single authority.

Benaloh & Kilian '86: split authority.

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

Setup:

\[ N = p \cdot q, \ p, q \text{ primes, } p + q \]
\[ + \text{prime with } +1 \text{ or } -1. \]
\[ KE_W \text{ public key of the authority,} \]
\[ K \text{ is not or \{ } <0, 1, \ldots, N-1 \text{ \}} \]
\[ \text{such that } K \neq n \text{ and the power:} \]
\[ \text{Kth residue: } K \neq 1 \]
\[ \text{Kth power: } K \neq x^y \text{ for all } x, \]
\[ \text{All other mod } p \text{ then } x^{p-1} = 1 \]
\[ \text{some do not have this for smaller } x. \]
\[ \text{Then } (a^y)^{p-1} = 1 \text{, go if } K \text{ is even or } K \text{ power then } K^{p-1} = 1. \text{ And vice versa.} \]
\[ \overline{K} (v_j, u_j) = K^{v_j} u_j \in \mathbb{Z}_N. \]

By raising this to \( p^{-1} \) the power we obtain:

\[ (K^{p^{-1}})^{v_j} \in \mathbb{Z}_N. \]

**Verifying stage (registration stage):**

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

Towards an authority.

**Verdicting stage**

Each vote submits one of her proposals.

**Tallying:**

The authority unchii-pies:

\[ K^{\sum v_j} (\prod u_j) \in \mathbb{Z}_N \]

and the decrypts:

\[ (K^{p^{-1}})^{\sum v_j} \in \mathbb{Z}_N \]

and derive \( \sum v_j \) from \( y \) by computing the discrete log using Baby step giant step.
Eligibility

Anonymity

As long as the authority is not corrupted, it's anonymous. Therefore, Benaloh & Young '86 split the authority, but still that scheme is not robust. So Benaloh '87 used a threshold secret sharing to make it more robust. With a (t, k)-threshold scheme we have k authorities and as long as t of them work properly, the decryption can be done.

Robustness

Accuracy & Verifiability

Fairness

Scaleability

None or less yes.

(Only possible problematic part is the tallying.)
Questions on schemes

- Describe
  - 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

\[ E_k(\text{token}) \rightarrow \text{Decryption mixnet} \rightarrow \text{token} \]

Voting stage

\[ E_k(\ldots) \rightarrow \text{Decryption mixnet} \rightarrow \text{token}, E_{K(\text{ID})}(\text{vote}) \]

\[ E_K(\text{plaintext}) = E_{K_1}(\ldots E_{K_5}(\text{plaintext}, r_3), \ldots, r_3) \quad E_{K_p}(h, r) = (g^r, K_x^p) \]
Chaum (1981)

Questions:
- Scalability?
- Security?
- Where do the secret keys come from?
- How is the signing done?

Sako & Kilian (1995)

Questions:
- Why do we need the registration stage?
  - Because of receipt freeness!

Baraloh & al. (1985, 86, 87) Hidden vote

Questions:
- Eligibility?
Sako & Kilian (1995)

Setup...

Registration stage

Voting stage

Tallying

Result
Chaum (1981)

Questions:
- Scalability?
- Security?
- Where do the secret keys come from?
- How is the signing done?

Sako & Kilian (1995)

Questions:
- Why do we need the registration stage?
- Is because of receipt freeness!

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

Questions:
- Eligibility?
Benaloh & al (1985, 86, 87)

Setup: \( N = p \cdot q + 1(p-1) \).

Pre-voting stage

Voting stage

Tallying stage

\[ \text{decrypt} \rightarrow \text{election result} \]
Chaum (1981)

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

Sako & Kilian (1995)

Questions:
- Why do we need the registration stage?
  - So because of receipt freeness!

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

Questions:
- Eligibility?
A practical secret voting scheme for large scale elections

Fujioka Okamoto Ohta '93

Voter

\[ \text{ballot} = \text{commitment}(\text{vote}, \text{key}) \]
\[ \text{msg} = \text{blinding}(\text{ballot}, \text{randomness}) \]

Admin

- checks
  - eligibility
  - one vote only
  - signature
- signs msg
- posts

\[ \{\text{ID, msg, voter's signature}\} \]

Counter

- check that ballot arrived on BB
- verify admin's signature

\[ \{\text{ballot, admin's signature}\} \]

\[ \{\text{sign(msg)}\} \]

\[ \{\text{vote}\} \]

\[ \text{verify signature} \]

\[ \{\text{result}\} \]

BB

Admin's signature invalid \( \rightarrow \) reveal ballot & signature

- More/less votes than counted \( \rightarrow \) reveal randomness

- Own ballot not on counter's list \( \rightarrow \) reveal ballot & admin's signature
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)

Verifiability: 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 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...
Kiayos & Yung (2002?)

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

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

$V_j$ gets $h \in G$ of size $q$
selects $a_j \in \mathbb{Z}_q^t$, publish $h_j = h^a_j$

2) Pre-Voting Stage

$V_i: n$ values $s_{i,j} \in \mathbb{Z}_q$

$V_i \xrightarrow{R_{i,j}} \ BBA$

BBA $\xrightarrow{R_j = \prod_{i=1}^{n} R_{i,j}} \ BBA$

3) Ballot-Casting

$V_j \xrightarrow{R_j = \prod_{i=1}^{n} s_{i,j}^4} \ BBA$

Vote: $V_j$

$t_j$ unknown to the voter

4) Tally

$T := \prod_{j=1}^{n} R_j = \prod_{j=1}^{n} R_{i,j}$

$T_j$
Hidden vote scheme

Eligibility: Check by signatures in the 1st 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.

Receivability: No, possible to proof which vote was given.

Robustness: No, but modification can solve this.

Questions: 1. is part? For obtaining \( h_{ij} \).
A Verifiable Multi-Authority Secret Election
Allowing Abstention from Voting.
By: Juang, Lei and Liaw.

**Initialization**
Counter publishes all parameters, and signs them.

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

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

**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.

**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).

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

**Announcement**
Counter publishes all accepted ballots.
eligibility - yes by using signatures
anonymity - ✓
verifiability - yes ✓
scalability - large scale
robust - mix-net
COERCION-RESISTANCE \(_{EC^2}\)
Juels, Catalano, Jakobsson

**Registration:**
\[
(\text{sk}_i, \text{pk}_i, \text{voter identifier}) \rightarrow (\text{sk}^*_i, \text{pk}^*_i)
\]

**Voting:**
\[
\begin{align*}
\text{E}_{1,1} & \rightarrow \text{E}_{1,2} \\
\text{E}_{2,1} & \rightarrow \text{E}_{2,2}
\end{align*}
\]

**Tallying:**
\[(\text{sk}_i, \text{pk}_i, \text{sk}^*_i) \rightarrow (x, p)\]

**Verifying:**
\[(\text{pk}_i, \text{pk}^*_i, x, p) \rightarrow \text{Verify} \rightarrow 0, 1, 3\]

Voter \(v_i\) (in choice \(a_i\) and credential \(c_i\)) for \(a_i \in \{0, 1\}\) sends through re-encryption mix net
\[
\text{E}_{1,1} = (g^a, g^b, \beta) = (g^{a_1}, g^{a_2}, g^{\cdot h^a_1})
\]
along with NIZK proofs \((\beta)\).
eligibility - yes by using signatures
anonymity - ✓
verifiability - yes ✓
scalability - large scale
robust - mix-net
Providing Receipt-Freeness in Mixnet-Based Voting Protocols

2004 (enhanced met of)
Sako & Kilian

Voter

TRR

Admin

Sig OKZ

Talliers
decrypt,
count

* (t,m) - threshold
El Gamal

TRR: Tamper Resistant Randomizer
DVRP: designated verifier re-encryption proof
Lee, Boyd, Dawson, ... 2004

(enhancement of Karo & Kilian)

Questions: No! :-)

Claim 2004

Questions: How it works?
Chaum 2004

- Chance to re-vote
- Vote in a booth
- One-time pad
- Control your vote

Supervised

Choose one

Compare to know that it's correctly posted

Step-by-step decryption

BB

BB plaintext
Lee, Boyd, Dawson, ... 2004
(enhancement of Kaho & Kilian)

Questions: No! :-)

Claim 2004

Questions: How it works?
Swiss (Geneva) Voting Scheme

**Registration:**
Postal letter with ID, PIN to every possible voter

**Voting:**
- **Login** with ID and PIN, birthday, city
- SSL encrypted
- Confirmation + date

**Voting Server** at Police Headquarters

**Encryption**
- \( E_{K1} K2 (\text{vote}) \) RSA

**Counting:**
- Mix Ballot Box
- Decrypt all votes with \( K1 \) \( K2 \)
- \( K1 \) \( K2 \) of different parties

**Results**
Estonian E-Voting System

General concept:

- Encrypted Vote
- Digital Signature
- System's public key
- System's private key
- Decryption
- E-Voters
- E-Votes

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 receives vote
3. VSS: Receives votes and separates signatures from encrypted votes
4. VCA: Receives only encrypted votes on CD (offline system) and performs counting of votes.
Considered systems, questions, left-overs


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 bulletin 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.


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.

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.

*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 $s_{j,i}$ for all voters such that these add up to 0, and sends exponentiated values $(g^{s_{j,i}}, h_{j,i}^{s_{j,i}})$ to the bulletin matrix. The bulletin board multiplies the columns: $R_j := \prod_i h_{j,i}^{s_{j,i}}$.

*Voting:* The voter $j$ raises $R_j$ to the $\alpha_j^{-1}$-th power and multiplies this with $f^{v_j}$, the value $B_j = h_{\sum_i s_{j,i}} f^{v_j}$ 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 destroyed 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.