University of Fribourg

Bern University of Applied Sciences

# Baloti: Verifiable E-voting for Foreign Residents of Switzerland

Oliver Spycher

SecVote Bertinoro / September 3rd, 2010

# Outline

# Outline

# PhD Workshop on E-voting

http://evotingphdworkshop2010.cased.de/
http://evotingphdworkshop2011.cased.de/

1. This fall in Fribourg Switzerland, Tuesday 07.09. - Wednesday 08.09.
2. Next spring in Germany / in English.
3. 30 - 45 Minute contributions.

Registration and more details:

- melanie.volkamer@cased.de
- oliver.spycher@bfh.ch

# Outline

PhD Workshop on E-voting

Motivation

SH10 Protocol - Modified Version

Selectio Helvetica for Baloti

# The Baloti Project

Baloti is an online platform.

- To familiarize foreign residents of Switzerland with the political environment.

# The Baloti Project

Baloti is an online platform.

- To familiarize foreign residents of Switzerland with the political environment.
- Funded by the integration fund of the Swiss Confederation.

# The Baloti Project

Baloti is an online platform.

- To familiarize foreign residents of Switzerland with the political environment.
- Funded by the integration fund of the Swiss Confederation.

www.baloti.ch

- Explains political processes.
- Informs on political issues and disputes.
- Offers participation in federal referendums.

# Participation

Users can cast their vote in federal referendums.

The ballot serves in a consultative function.

# Participation

Users can cast their vote in federal referendums.

The ballot serves in a consultative function.

- The referendum is explained in 14 languages.

# Participation

Users can cast their vote in federal referendums.

The ballot serves in a consultative function.

- The referendum is explained in 14 languages.
- Pros and Cons are explained in 14 languages.

# Participation

Users can cast their vote in federal referendums.

The ballot serves in a consultative function.

- The referendum is explained in 14 languages.
- Pros and Cons are explained in 14 languages.
- The significance of privacy and integrity is explained in 14 languages.

# Trust

The Baloti project is carried out by an interdiscipinary team at the ZDA (Zentrum für Demokratie) in Aarau.

They are well aware of the reservation immigrants can have towards political institutions.

Baloti runs Selectio Helvetica

- A trustworthy, transparent E-voting system.
- To create justified trust among its users.

# Outline

# Electronic Channel for Hybrid Systems

- Many governments aim at integrating a new e-voting channel with their traditional paper-based channel.

# Electronic Channel for Hybrid Systems

- Many governments aim at integrating a new e-voting channel with their traditional paper-based channel.
- Integration as a *Hybrid System* aims at coercion-resistance.

→ Revoke e-vote and replace it at polling-station.

# Electronic Channel for Hybrid Systems

- Many governments aim at integrating a new e-voting channel with their traditional paper-based channel.
- Integration as a *Hybrid System* aims at coercion-resistance.

→ Revoke e-vote and replace it at polling-station.

## Requirements on Electronic Channel

- Proof of eligibility.
- Proof of ownership.
- Encryption function applied on votes must allow re-encryption.
- Encryption function applied on votes must allow proof of correct re-encryption.

# Setup a PKI per Voter for DSA

Voters are assigned their

- *private key* $s \mod q$
- *public key* $S = g^s \mod p \qquad (p = 2q + 1)$

# Setup a PKI per Voter for DSA

Voters are assigned their

- *private key* $s \mod q$
- *public key* $S = g^s \mod p \qquad (p = 2q + 1)$

Voters can prove that they know the private key that matches their public key. (Zero-Knowledge Proof)

# Setup a PKI per Voter for DSA

Voters are assigned their

- *private key* $s \mod q$
- *public key* $S = g^s \mod p$ $\qquad (p = 2q + 1)$

Voters can prove that they know the private key that matches their public key. (Zero-Knowledge Proof)

## Distribution

Voting Officials jointly create and publish the public key and secretly reveal their share of the private key to the Voter.

# Setup a PKI per Voter for DSA

Voters are assigned their

- *private key* $s \mod q$
- *public key* $S = g^s \mod p \qquad (p = 2q + 1)$

Voters can prove that they know the private key that matches their public key. (Zero-Knowledge Proof)

## Distribution

Voting Officials jointly create and publish the public key and secretly reveal their share of the private key to the Voter.
→ R. Gennaro, Jarecki, Krawczyk, T. Rabin: Eurocrypt 1999

# Setup a PKI per Voter for DSA

Voters are assigned their

- *private key* $s \mod q$
- *public key* $S = g^s \mod p \qquad (p = 2q + 1)$

Voters can prove that they know the private key that matches their public key. (Zero-Knowledge Proof)

## Distribution

Voting Officials jointly create and publish the public key and secretly reveal their share of the private key to the Voter.
$\rightarrow$ R. Gennaro, Jarecki, Krawczyk, T. Rabin: Eurocrypt 1999

This only has to be done once.

# A First Naive Approach wihout Privacy

This is the public bulletin board.

| **Voter Roll** | | | |
|---|---|---|---|
| 1: Hugo | | | |
| 2: Mark | | | |
| 3: Peter | | | |

# A First Naive Approach wihout Privacy

This is the public bulletin board.

| **Voter Roll** | **Public** | | |
|---|---|---|---|
| 1: Hugo | $S_1 = g^{s_1}$ | | |
| 2: Mark | $S_2 = g^{s_2}$ | | |
| 3: Peter | $S_3 = g^{s_3}$ | | |

:

# A First Naive Approach wihout Privacy

This is the public bulletin board.

| **Voter Roll** | **Public** | **Encryption of Vote** | |
|---|---|---|---|
| 1: Hugo | $S_1 = g^{s_1}$ | $w_1 = (h^{k_1}, yes \cdot e^{k_1})$ | |
| 2: Mark | $S_2 = g^{s_2}$ | $w_2 = (h^{k_2}, yes \cdot e^{k_2})$ | |
| 3: Peter | $S_3 = g^{s_3}$ | $w_3 = (h^{k_3}, yes \cdot e^{k_3})$ | |

:

# A First Naive Approach wihout Privacy

This is the public bulletin board.

| Voter Roll | Public | Encryption of Vote | Signature of Enc |
|---|---|---|---|
| 1: Hugo | $S_1 = g^{s_1}$ | $w_1 = (h^{k_1}, yes \cdot e^{k_1})$ | $sign(w_1, s_1, g, q)$ |
| 2: Mark | $S_2 = g^{s_2}$ | $w_2 = (h^{k_2}, yes \cdot e^{k_2})$ | $sign(w_2, s_2, g, q)$ |
| 3: Peter | $S_3 = g^{s_3}$ | $w_3 = (h^{k_3}, yes \cdot e^{k_3})$ | $sign(w_3, s_3, g, q)$ |

:

# A First Naive Approach wihout Privacy

This is the public bulletin board.

| Voter Roll | Public | Encryption of Vote | Signature of Enc |
|---|---|---|---|
| 1: Hugo | $S_1 = g^{s_1}$ | $w_1 = (h^{k_1}, yes \cdot e^{k_1})$ | $sign(w_1, s_1, g, q)$ |
| 2: Mark | $S_2 = g^{s_2}$ | $w_2 = (h^{k_2}, yes \cdot e^{k_2})$ | $sign(w_2, s_2, g, q)$ |
| 3: Peter | $S_3 = g^{s_3}$ | $w_3 = (h^{k_3}, yes \cdot e^{k_3})$ | $sign(w_3, s_3, g, q)$ |

**Proof of Eligibility**: Simple

:

# A First Naive Approach wihout Privacy

This is the public bulletin board.

| Voter Roll | Public | Encryption of Vote | Signature of Enc |
|---|---|---|---|
| 1: Hugo | $S_1 = g^{s_1}$ | $w_1 = (h^{k_1}, yes \cdot e^{k_1})$ | $sign(w_1, s_1, g, q)$ |
| 2: Mark | $S_2 = g^{s_2}$ | $w_2 = (h^{k_2}, yes \cdot e^{k_2})$ | $sign(w_2, s_2, g, q)$ |
| 3: Peter | $S_3 = g^{s_3}$ | $w_3 = (h^{k_3}, yes \cdot e^{k_3})$ | $sign(w_3, s_3, g, q)$ |

**Proof of Eligibility**: Simple
**Proof of Ownership**: Simple

:

# A First Naive Approach wihout Privacy

This is the public bulletin board.

| Voter Roll | Public | Encryption of Vote | Signature of Enc |
|---|---|---|---|
| 1: Hugo | $S_1 = g^{s_1}$ | $w_1 = (h^{k_1}, yes \cdot e^{k_1})$ | $sign(w_1, s_1, g, q)$ |
| 2: Mark | $S_2 = g^{s_2}$ | $w_2 = (h^{k_2}, yes \cdot e^{k_2})$ | $sign(w_2, s_2, g, q)$ |
| 3: Peter | $S_3 = g^{s_3}$ | $w_3 = (h^{k_3}, yes \cdot e^{k_3})$ | $sign(w_3, s_3, g, q)$ |

**Proof of Eligibility**: Simple
**Proof of Ownership**: Simple
**Hugo needs to revoke his vote before casting a paper vote**:

# A First Naive Approach wihout Privacy

This is the public bulletin board.

| Voter Roll | Public | Encryption of Vote | Signature of Enc |
|---|---|---|---|
| 1: Hugo | $S_1 = g^{s_1}$ | $w_1 = (h^{k_1}, yes \cdot e^{k_1})$ | $sign(w_1, s_1, g, q)$ |
| 2: Mark | $S_2 = g^{s_2}$ | $w_2 = (h^{k_2}, yes \cdot e^{k_2})$ | $sign(w_2, s_2, g, q)$ |
| 3: Peter | $S_3 = g^{s_3}$ | $w_3 = (h^{k_3}, yes \cdot e^{k_3})$ | $sign(w_3, s_3, g, q)$ |

**Proof of Eligibility**: Simple
**Proof of Ownership**: Simple
**Hugo needs to revoke his vote before casting a paper vote**:

1. Choose uniformly random $z$ from $[1, .., q]$

# A First Naive Approach wihout Privacy

This is the public bulletin board.

| Voter Roll | Public | Encryption of Vote | Signature of Enc |
|---|---|---|---|
| 1: Hugo | $S_1 = g^{s_1}$ | $w_1 = (h^{k_1}, yes \cdot e^{k_1})$ | $sign(w_1, s_1, g, q)$ |
| 2: Mark | $S_2 = g^{s_2}$ | $w_2 = (h^{k_2}, yes \cdot e^{k_2})$ | $sign(w_2, s_2, g, q)$ |
| 3: Peter | $S_3 = g^{s_3}$ | $w_3 = (h^{k_3}, yes \cdot e^{k_3})$ | $sign(w_3, s_3, g, q)$ |

**Proof of Eligibility**: Simple

**Proof of Ownership**: Simple

**Hugo needs to revoke his vote before casting a paper vote**:

1. Choose uniformly random $z$ from $[1, .., q]$
2. Compute $Re\text{-}Enc(w_1, z) = (h^{k_1} \cdot h^z, yes \cdot e^{k_1} \cdot e^z)$ and proof.

# A First Naive Approach wihout Privacy

This is the public bulletin board.

| Voter Roll | Public | Encryption of Vote | Signature of Enc |
|---|---|---|---|
| 1: Hugo | $S_1 = g^{s_1}$ | $w_1 = (h^{k_1}, yes \cdot e^{k_1})$ | $sign(w_1, s_1, g, q)$ |
| 2: Mark | $S_2 = g^{s_2}$ | $w_2 = (h^{k_2}, yes \cdot e^{k_2})$ | $sign(w_2, s_2, g, q)$ |
| 3: Peter | $S_3 = g^{s_3}$ | $w_3 = (h^{k_3}, yes \cdot e^{k_3})$ | $sign(w_3, s_3, g, q)$ |

**Proof of Eligibility**: Simple
**Proof of Ownership**: Simple
**Hugo needs to revoke his vote before casting a paper vote**:

1. Choose uniformly random $z$ from $[1, .., q]$
2. Compute $Re\text{-}Enc(w_1, z) = (h^{k_1} \cdot h^z, yes \cdot e^{k_1} \cdot e^z)$ and proof.
3. Have polling-station authorities sign both.

# A First Naive Approach wihout Privacy

This is the public bulletin board.

| Voter Roll | Public | Encryption of Vote | Signature of Enc |
|---|---|---|---|
| 1: Hugo | $S_1 = g^{s_1}$ | $w_1 = (h^{k_1}, yes \cdot e^{k_1})$ | $sign(w_1, s_1, g, q)$ |
| 2: Mark | $S_2 = g^{s_2}$ | $w_2 = (h^{k_2}, yes \cdot e^{k_2})$ | $sign(w_2, s_2, g, q)$ |
| 3: Peter | $S_3 = g^{s_3}$ | $w_3 = (h^{k_3}, yes \cdot e^{k_3})$ | $sign(w_3, s_3, g, q)$ |

**Proof of Eligibility**: Simple
**Proof of Ownership**: Simple
**Hugo needs to revoke his vote before casting a paper vote**:

1. Choose uniformly random $z$ from $[1, .., q]$
2. Compute $Re\text{-}Enc(w_1, z) = (h^{k_1} \cdot h^z, yes \cdot e^{k_1} \cdot e^z)$ and proof.
3. Have polling-station authorities sign both.
4. Cast $Re\text{-}Enc(w_1, z)$, proof and signature to revocation board.

# A First Naive Approach wihout Privacy

This is the public bulletin board.

| Voter Roll | Public | Encryption of Vote | Signature of Enc |
|---|---|---|---|
| 1: Hugo | $S_1 = g^{s_1}$ | $w_1 = (h^{k_1}, yes \cdot e^{k_1})$ | $sign(w_1, s_1, g, q)$ |
| 2: Mark | $S_2 = g^{s_2}$ | $w_2 = (h^{k_2}, yes \cdot e^{k_2})$ | $sign(w_2, s_2, g, q)$ |
| 3: Peter | $S_3 = g^{s_3}$ | $w_3 = (h^{k_3}, yes \cdot e^{k_3})$ | $sign(w_3, s_3, g, q)$ |

**Proof of Eligibility**: Simple
**Proof of Ownership**: Simple
**Hugo needs to revoke his vote before casting a paper vote**:

1. Choose uniformly random $z$ from $[1, .., q]$
2. Compute $Re\text{-}Enc(w_1, z) = (h^{k_1} \cdot h^z, yes \cdot e^{k_1} \cdot e^z)$ and proof.
3. Have polling-station authorities sign both.
4. Cast $Re\text{-}Enc(w_1, z)$, proof and signature to revocation board.

So what about Privacy?

# Introduce Pseudonyms for Privacy

Mixing authorities jointly compute Pseudonyms.

1. Select random $\alpha$ from $\mathbb{Z}_q$

| Voter Roll | Public |
|---|---|
| 1: Hugo | $S_1 = g^{s_1}$ |
| 2: Mark | $S_2 = g^{s_2}$ |
| 3: Peter | $S_3 = g^{s_3}$ |

This is the voting board:

| Pseudonym | | |
|---|---|---|
| | | |
| | | |
| | | |

# Introduce Pseudonyms for Privacy

Mixing authorities jointly compute Pseudonyms.

1. Select random $\alpha$ from $\mathbb{Z}_q$
2. Publish $\hat{g} = g^{\alpha} \mod p$ (Pseudonym Generator)

| Voter Roll | Public |
|---|---|
| 1: Hugo | $S_1 = g^{s_1}$ |
| 2: Mark | $S_2 = g^{s_2}$ |
| 3: Peter | $S_3 = g^{s_3}$ |

This is the voting board:

| Pseudonym | | |
|---|---|---|
| | | |
| | | |
| | | |

# Introduce Pseudonyms for Privacy

Mixing authorities jointly compute Pseudonyms.

1. Select random $\alpha$ from $\mathbb{Z}_q$
2. Publish $\hat{g} = g^{\alpha} \mod p$ (Pseudonym Generator)
3. Compute Pseudonym $\hat{S}_{\pi(i)} = S_i^{\alpha} (= \hat{g}^{s_i})$

| Voter Roll | Public |
|---|---|
| 1: Hugo | $S_1 = g^{s_1}$ |
| 2: Mark | $S_2 = g^{s_2}$ |
| 3: Peter | $S_3 = g^{s_3}$ |

This is the voting board:

| Pseudonym | | |
|---|---|---|
| | | |
| | | |
| | | |

# Introduce Pseudonyms for Privacy

Mixing authorities jointly compute Pseudonyms.

1. Select random $\alpha$ from $\mathbb{Z}_q$
2. Publish $\hat{g} = g^{\alpha} \mod p$ (Pseudonym Generator)
3. Compute Pseudonym $\hat{S}_{\pi(i)} = S_i^{\alpha} (= \hat{g}^{s_i})$

| Voter Roll | Public |
|---|---|
| 1: Hugo | $S_1 = g^{s_1}$ |
| 2: Mark | $S_2 = g^{s_2}$ |
| 3: Peter | $S_3 = g^{s_3}$ |

This is the voting board:

| Pseudonym | | |
|---|---|---|
| | | |
| | | |
| | | |

# Introduce Pseudonyms for Privacy

Mixing authorities jointly compute Pseudonyms.

1. Select random $\alpha$ from $\mathbb{Z}_q$
2. Publish $\hat{g} = g^{\alpha} \mod p$ (Pseudonym Generator)
3. Compute Pseudonym $\hat{S}_{\pi(i)} = S_i^{\alpha} (= \hat{g}^{s_i})$

| Voter Roll | Public |
|---|---|
| 1: Hugo | $S_1 = g^{s_1}$ |
| 2: Mark | $S_2 = g^{s_2}$ |
| 3: Peter | $S_3 = g^{s_3}$ |

This is the voting board:

| Pseudonym | | |
|---|---|---|
| | | |
| | | |
| | | |

# Introduce Pseudonyms for Privacy

Mixing authorities jointly compute Pseudonyms.

1. Select random $\alpha$ from $\mathbb{Z}_q$
2. Publish $\hat{g} = g^{\alpha} \mod p$ (Pseudonym Generator)
3. Compute Pseudonym $\hat{S}_{\pi(i)} = S_i^{\alpha} (= \hat{g}^{s_i})$

| **Voter Roll** | **Public** |
|---|---|
| 1: Hugo | $S_1 = g^{s_1}$ |
| 2: Mark | $S_2 = g^{s_2}$ |
| 3: Peter | $S_3 = g^{s_3}$ |

This is the voting board:

| **Pseudonym** | | |
|---|---|---|
| $\hat{S}_1 = \hat{g}^{s_2}$ | | |
| $\hat{S}_2 = \hat{g}^{s_3}$ | | |
| $\hat{S}_3 = \hat{g}^{s_1}$ | | |

# Introduce Pseudonyms for Privacy

Mixing authorities jointly compute Pseudonyms.

1. Select random $\alpha$ from $\mathbb{Z}_q$
2. Publish $\hat{g} = g^{\alpha} \mod p$ (Pseudonym Generator)
3. Compute Pseudonym $\hat{S}_{\pi(i)} = S_i^{\alpha} (= \hat{g}^{s_i})$

| Voter Roll | Public |
|---|---|
| 1: Hugo | $S_1 = g^{s_1}$ |
| 2: Mark | $S_2 = g^{s_2}$ |
| 3: Peter | $S_3 = g^{s_3}$ |

This is the voting board:

| Pseudonym | Encryption of Vote | |
|---|---|---|
| $\hat{S}_1 = \hat{g}^{s_2}$ | $w_1 = (h^{k_1}, yes \cdot e^{k_1})$ | |
| $\hat{S}_2 = \hat{g}^{s_3}$ | $w_2 = (h^{k_2}, yes \cdot e^{k_2})$ | |
| $\hat{S}_3 = \hat{g}^{s_1}$ | $w_3 = (h^{k_3}, yes \cdot e^{k_3})$ | |

# Introduce Pseudonyms for Privacy

Mixing authorities jointly compute Pseudonyms.

1. Select random $\alpha$ from $\mathbb{Z}_q$
2. Publish $\hat{g} = g^{\alpha} \mod p$ (Pseudonym Generator)
3. Compute Pseudonym $\hat{S}_{\pi(i)} = S_i^{\alpha} (= \hat{g}^{s_i})$

| Voter Roll | Public |
|---|---|
| 1: Hugo | $S_1 = g^{s_1}$ |
| 2: Mark | $S_2 = g^{s_2}$ |
| 3: Peter | $S_3 = g^{s_3}$ |

This is the voting board:

| Pseudonym | Encryption of Vote | Signature of Enc |
|---|---|---|
| $\hat{S}_1 = \hat{g}^{s_2}$ | $w_1 = (h^{k_1}, yes \cdot e^{k_1})$ | $sign(w_1, s_2, \hat{g}, q)$ |
| $\hat{S}_2 = \hat{g}^{s_3}$ | $w_2 = (h^{k_2}, yes \cdot e^{k_2})$ | $sign(w_2, s_3, \hat{g}, q)$ |
| $\hat{S}_3 = \hat{g}^{s_1}$ | $w_3 = (h^{k_3}, yes \cdot e^{k_3})$ | $sign(w_3, s_1, \hat{g}, q)$ |

# Revocation

| Voter Roll | Public |
|---|---|
| 1: Hugo | $S_1 = g^{s_1}$ |
| 2: Mark | $S_2 = g^{s_2}$ |
| 3: Peter | $S_3 = g^{s_3}$ |

| Pseudonym | Encryption of Vote | Signature of Enc |
|---|---|---|
| $\hat{S}_1 = \hat{g}^{s_2}$ | $w_1 = (h^{k_1}, yes \cdot e^{k_1})$ | $sign(w_1, s_2, \hat{g}, q)$ |
| $\hat{S}_2 = \hat{g}^{s_3}$ | $w_2 = (h^{k_2}, yes \cdot e^{k_2})$ | $sign(w_2, s_3, \hat{g}, q)$ |
| $\hat{S}_3 = \hat{g}^{s_1}$ | $w_3 = (h^{k_3}, yes \cdot e^{k_3})$ | $sign(w_3, s_1, \hat{g}, q)$ |

**Proof of Eligibility**:

1. Hugo reveals his Pseudonym $\hat{S}_3$.
2. He proves $ZKP[(s_1) : S_1 = g^{s_1} \wedge \hat{S}_3 = \hat{g}^{s_1}]$

**Proof of Ownership**: Simple

**Revoke encrypted vote** $w_3$: Same as in naive version.

# Properties

1. Individual Verifiability (Public board and ElGamal randomness)
2. Universal Verifiability (Public board Authorities reproduce and publish their private key for vote decryption)
3. Privacy (even as to whether voters participate)
4. Verifiability of Eligibility (Authorities prove correct pseudonym generation)
5. Integrity / Accuracy (Public Board)
6. Authenticated Channel needed only once (at key generation)
7. Coercion / Vote Buying attacks are mitigated by allowing revocation (→ Hybrid System)

# Outline

PhD Workshop on E-voting

Motivation

SH10 Protocol - Modified Version

Selectio Helvetica for Baloti

# Baloti Specific Requirements

Selectio Helvetica is meant to give the experience of a verifiable voting system that could be used for governmental votes.

→ We do not change the protocol

→ We extend the protocol to meet the Baloti specific requirements

# Baloti Specific Requirements

Selectio Helvetica is meant to give the experience of a verifiable voting system that could be used for governmental votes.

→ We do not change the protocol

→ We extend the protocol to meet the Baloti specific requirements

## Baloti Requirements

1. Web-browser on client side
2. Users cannot memorize long, unintuitive values. (e.g. their private key)
3. Users can memorize a password-like value
4. Users can join the voter-roll at any time and instantly vote
5. A voter-roll entry is an email address.

# Selectio Helvetica - Outline 1

Voters need a password-like voting-code for casting votes and individual verifiability.

# Selectio Helvetica - Outline 1

Voters need a password-like voting-code for casting votes and individual verifiability.

## Registration

1. Baloti grants a user the right to vote, signs his email address, sends both to Selectio Helvetica
2. Selectio Helvetica sends a registration credential to voter.
3. Voter chooses his voting-code and sends one Shamir share each to authorities $A_i$ along with the registration credential.
4. Authority $A_i$ maps the share of the voting-code to a share of the DSA private key.

# Selectio Helvetica - Outline 2

## Vote Casting

Voter makes his choice in the browser, enters his voting-code and clicks *cast vote*.

1. The browser sends each $A_i$ its share of the voting-code.
2. Each $A_i$ returns its share of the mapped private key $s$.
3. Voter reconstructs his private key $s$. (Shamir)

For instant *individual verifiability*, the voter shares the randomness $k$ used in the ElGamal encryption of the vote among multiple authorities. (Use the voting-code to re-obtain.)

# Selectio Helvetica - Properties

## Voters with a Good Memory

Assuming secure platform, and correct code running in browser, the properties of SH10 can almost be met.

1. The email provider and the sending authority could steal a voter's registration credential.
2. However, the voter would notice.

## Forgetful Voters

Voter who forgets his voting-code is punished.

1. He can ask the authorities to send their shares of the voting-code by email.
2. Voter opens each email, and copy-pastes each share into the browser. The browser computes the original voting-code.