University of Fribourg

Bern University of Applied Sciences

Selectio Helvetica

A Verifiable Internet Voting System

Oliver Spycher

Krems, May 5th, 2011

Outline

SH Project

SH Protocol

SH System as in Baloti

Outline

SH Project

SH Protocol

SH System as in Baloti

Internet Voting and the SH Project

Perfect internet voting has not been invented

SH to Address Trust

- SH offers a verifiable internet voting service to vote organizers
- SH publishes all documentation and exposes inherent security concerns
- seeks to raise debates on security among all stakeholders, not just security experts

The Baloti Project

- is conducted by our partner institute ZDA
- offers vote participation to migrant population of CH
- uses the SH *light* service

Trust in the Integrity of a Vote

Integrity means that

- all legitimate votes are counted as cast
- only legitimate votes are counted

→ How relevant is trustworthiness?

→ When would you trust your polling station crew?

→ What about internet voting?

Verify Integrity

Verifiability is covered by

- Individual Verifiability: Your vote reached the ballot box
- Eligibility Verifiability: All votes in the ballot box are legitimate
- Universal Verifiability: All votes from the ballot box have been counted

→ But what about secrecy?

Outline

Outline

SH Project

SH Protocol

SH System as in Baloti

Introduce a Public Board

Voter Roll	
1: Angela	
2: Nick	
3: Silvio	

Introduce a Public Board

Voter Roll	Vote
1: Angela	*yes*
2: Nick	*yes*
3: Silvio	*yes*

Introduce a Public Board

Voter Roll	Vote
1: Angela	*yes*
2: Nick	*yes*
3: Silvio	*yes*

Verifiability

- Individual
- Eligibility - no
- Universal

A First Naive Approach without Secrecy I

Keys for Signing Votes (DSA over safe primes)

- private key s_i
- public key S_i, g $(S_i = g^{s_i})$

Use S_i and g to verify signature $sign(m, s_i, g)$ of m

Can't compute s_i given S_i or any other public values

$\rightarrow$ If s_i is kept secret and m is a vote, the vote must originate from an eligible voter

A First Naive Approach without Secrecy II

Voter Roll	Public		
1: Angela	$S_1 = g^{s_1}$		
2: Nick	$S_2 = g^{s_2}$		
3: Silvio	$S_3 = g^{s_3}$		

A First Naive Approach without Secrecy II

Voter Roll	Public	Vote	Signature of Vote
1: Angela	$S_1 = g^{s_1}$	*yes*	$sign(yes, s_1, g)$
2: Nick	$S_2 = g^{s_2}$	*yes*	$sign(yes, s_2, g)$
3: Silvio	$S_3 = g^{s_3}$	*yes*	$sign(yes, s_3, g)$

Verifiability

- Individual
- Eligibility
- Universal

Introducing Secrecy

Secrecy Requirements

1. Privacy (no link vote - voter)
2. Fairness (no premature result)

Introducing Secrecy

Secrecy Requirements

1. Privacy (no link vote - voter)
2. Fairness (no premature result)

Step by Step

- cast encrypted votes (fairness if trustworthy authorities)
- use pseudonyms for signing (secrecy if trustworthy authorities)
- separation of duty (secrecy and easier to trust authorities)

Cast Encrypted Votes I

Keys for Encrypting Votes (IND-CPA ElGamal)

- private key d
- public key e, h $\qquad e = h^d$

Use d to decrypt encryption $enc(m, e, h)$ of m

Can't decrypt messages without d (or randomness)

Can't compute d given e or any public values

Cast Encrypted Votes II

Use public key e to encrypt votes

Voter Roll	Public		
1: Angela	$S_1 = g^{s_1}$		
2: Nick	$S_2 = g^{s_2}$		
3: Silvio	$S_3 = g^{s_3}$		

Cast Encrypted Votes II

Use public key e to encrypt votes

Voter Roll	Public	Encrypted Vote	Signature of Vote
1: Angela	$S_1 = g^{s_1}$	$w_1 = enc(yes, e, h)$	$sign(w_1, s_1, g)$
2: Nick	$S_2 = g^{s_2}$	$w_2 = enc(yes, e, h)$	$sign(w_2, s_2, g)$
3: Silvio	$S_3 = g^{s_3}$	$w_3 = enc(yes, e, h)$	$sign(w_3, s_3, g)$

Verifiability

- Individual
- Eligibility
- Universal (After the voting phase, d is published)

Use Pseudonyms for Signing I

Produce Pseudonyms

- For public key S_i select a new random distinct index j.
- Publish pseudonym $\hat{S}_j$ as S_i^{α} and $\hat{g}$ as g^{α}. (α secret)

Can't link any $\hat{S}_j$ to S_i given all public values

Use s_i and $\hat{g}$ to compute own pseudonym $\hat{S}_j$ as $\hat{g}^{s_i}$

because $\hat{g}^{s_i} = (g^{\alpha})^{s_i} = g^{\alpha \cdot s_i} = g^{s_i \cdot \alpha} = (g^{s_i})^{\alpha} = S_i^{\alpha} = \hat{S}_j$

Use $\hat{S}_i$ and $\hat{g}$ to verify signature $sign(m, s_i, \hat{g})$ of m

Use Pseudonyms for Signing II

Compute signature using s_i and $\hat{g}$.

Voter Roll	Public
1: Angela	$S_1 = g^{s_1}$
2: Nick	$S_2 = g^{s_2}$
3: Silvio	$S_3 = g^{s_3}$

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

Use Pseudonyms for Signing II

Compute signature using s_i and $\hat{g}$.

Voter Roll	Public
1: Angela	$S_1 = g^{s_1}$
2: Nick	$S_2 = g^{s_2}$
3: Silvio	$S_3 = g^{s_3}$

Pseudonym	Encryption of Vote	Signature of Enc
$\hat{S}_1 = \hat{g}^{s_2}$	$w_1 = enc(yes, e, h)$	$sign(w_1, s_2, \hat{g})$
$\hat{S}_2 = \hat{g}^{s_3}$	$w_2 = enc(yes, e, h)$	$sign(w_2, s_3, \hat{g})$
$\hat{S}_3 = \hat{g}^{s_1}$	$w_3 = enc(yes, e, h)$	$sign(w_3, s_1, \hat{g})$

Use Pseudonyms for Signing II

Compute signature using s_i and $\hat{g}$.

Voter Roll	Public
1: Angela	$S_1 = g^{s_1}$
2: Nick	$S_2 = g^{s_2}$
3: Silvio	$S_3 = g^{s_3}$

Pseudonym	Encryption of Vote	Signature of Enc
$\hat{S}_1 = \hat{g}^{s_2}$	$w_1 = enc(yes, e, h)$	$sign(w_1, s_2, \hat{g})$
$\hat{S}_2 = \hat{g}^{s_3}$	$w_2 = enc(yes, e, h)$	$sign(w_2, s_3, \hat{g})$
$\hat{S}_3 = \hat{g}^{s_1}$	$w_3 = enc(yes, e, h)$	$sign(w_3, s_1, \hat{g})$

Verifiability

- Individual (Compute pseudonym to locate vote)
- Eligibility (ZKP of mix)
- Universal (After the voting phase, d is published)

Separation of Duty I

Distribute Tasks Among Multiple Trustees

- Distribute d among trustees
  - → published: $e_1 = h^{d_1}, e_2 = h^{d_2}, ..., e_n = h^{d_n}$
  - → public key e computed as $e_1 \cdot e_2 \cdots e_n$
  - → private key d computed as $d_1 + d_2 + ... + d_n$
  - → can't compute d, unless all $d_1, d_2, ..., d_n$ are known
- Have trustees iteratively perform pseudonym generation
  - → secret $\alpha = \alpha_1 \cdot \alpha_2 \cdots \alpha_n$
  - → can't compute α, unless all $\alpha_1, \alpha_2, ..., \alpha_n$ are known

---

→ Secrecy preserved unless *all* trustees collude

Separation of Duty II

No need to trust single entity

Voter Roll	Public
1: Angela	$S_1 = g^{s_1}$
2: Nick	$S_2 = g^{s_2}$
3: Silvio	$S_3 = g^{s_3}$

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

Separation of Duty II

No need to trust single entity

Voter Roll	Public
1: Angela	$S_1 = g^{s_1}$
2: Nick	$S_2 = g^{s_2}$
3: Silvio	$S_3 = g^{s_3}$

Pseudonym	Encryption of Vote	Signature of Enc
$\hat{S}_1 = \hat{g}^{s_2}$	$w_1 = enc(yes, e, h)$	$sign(w_1, s_2, \hat{g})$
$\hat{S}_2 = \hat{g}^{s_3}$	$w_2 = enc(yes, e, h)$	$sign(w_2, s_3, \hat{g})$
$\hat{S}_3 = \hat{g}^{s_1}$	$w_3 = enc(yes, e, h)$	$sign(w_3, s_1, \hat{g})$

Separation of Duty II

No need to trust single entity

Voter Roll	Public
1: Angela	$S_1 = g^{s_1}$
2: Nick	$S_2 = g^{s_2}$
3: Silvio	$S_3 = g^{s_3}$

Pseudonym	Encryption of Vote	Signature of Enc
$\hat{S}_1 = \hat{g}^{s_2}$	$w_1 = enc(yes, e, h)$	$sign(w_1, s_2, \hat{g})$
$\hat{S}_2 = \hat{g}^{s_3}$	$w_2 = enc(yes, e, h)$	$sign(w_2, s_3, \hat{g})$
$\hat{S}_3 = \hat{g}^{s_1}$	$w_3 = enc(yes, e, h)$	$sign(w_3, s_1, \hat{g})$

Verifiability

- Individual (Compute pseudonym to locate vote)
- Eligibility (ZKP of mix)
- Universal (After the voting phase, d is published)

Features

- Verifiability with no trust constraints towards authorities
- Secrecy assuming at least 1 trustworthy authority
- Privacy in participation as an additional secrecy feature
- Re-usable credentials (personal authentication only once)
- Revocability at polling station despite privacy in participation
- No mixing of votes required before decrypting (fast results)

However..

Points of Debate

- Handing out secret key s_i to friends
- Handing out secret key s_i to vote-buyers or coercers
- Long-term privacy
- Voter's platform (computer, voting program)
- Anonymous channel (hard to implement)
- Disputes

Outline

SH Project

SH Protocol

SH System as in Baloti

Special Constraint: Evolving Voter Roll

Voters can join the voter roll anytime (Baloti)

Solution

- Key-pairs (S_i, s_i) are generated by trustees (separation of duty)
- Voter informs vote organizer that he wants to participate
- Vote organizer sends email address and signature of approval to SH
- SH sends registration credential to voter (link) by email
- Voter clicks on link, chooses password and sends a distinct hash to each trustee
- Each trustee associates email address and hash of password with its share of s_i

$\longrightarrow$ Voters obtain their secret s_i by entering their password.

Limitations towards Protocol

- Quality of voter roll depends on authentication of email addresses
- Privacy in participation as an additional secrecy feature is limited for the benefit of usability

Protocol Limitations not Inherent to Baloti

- Handing out secret key s_i to friends
- Handing out secret key s_i to vote-buyers or coercers

Thank You!

Questions / Remarks

..go cast your vote at **www.baloti.ch**