Verifiable Student Board Elections with UniVote

Rolf Haenni & Reto E. Koenig
April 4th, 2014
UniVote: Project Overview
Project Overview

- UniVote = Internet voting system for student board elections at Swiss universities
- 13 months development (February 2012 – February 2013)
  - 1 main developer and server administrator (50% assistant)
  - 1 PhD student (25% developer for UniVote)
  - 4 professors (protocols, specification, system design, …)
- Current version: 1.7
- Source code and documentation publicly available: https://www.univote.ch/documentation
- Verification software available (independent student project)
- Voting simulator under development (student project at HSR)
Previous and Future Elections

- Complex ballots with party lists (similar to NR elections)
- Previous elections
  - March 2013: University of Bern (11’000)
  - April 2013: Bern University of Applied Sciences (6’000)
  - May 2013: University of Zürich (26’000)
  - September 2013: University of Lucerne (3’000)
  - October 2013: Best Teacher Award, University of Lucerne
- Current and future elections
  - April 2014: Bern University of Applied Sciences
  - October 2014: Best Teacher Award, University of Lucerne
  - Elections in 2015: UniBE, UniZH, UniLU
- Average participation: $\approx 10\%$
UniVote User Interface

VSBFH Studierendenratswahl 2014

Key Entry ➔ Vote ➔ Confirmation

Please prepare your vote by dragging the preferred list and candidates from the left column to the ballot on the right-hand-side. You can cast the ballot whenever you are ready.

Candidates

List 1

SHEPPS

List 2

Kaufmann Claudia
Kaufmann Claudia

List 3

Dimitrejvic Jelena
Dimitrejvic Jelena

List 4

Zurlinden Patrik
Zurlinden Patrik

List 5

Matter Celine
Matter Celine

List 6

Martin Lina
Martin Lina
Zimmermann Jessica

Your Selection

List 4

SHEPPS

Buri Samuel
Marwik Darius
Sommer Michael
Lüdi Marius
Schwendimann Adrian
Willi Benjamin
Käser Phillip
System Properties and Design
Verifiability

“One should verify the election, not the election system.”

Ben Adida

- Individual verifiability: Correctness and inclusion of single vote
- Universal verifiability: Correctness of final election result
System Properties

- PKI based on existing Swiss university eID infrastructure
- Individually and universally verifiable
- Public election board (EB)
  - All election data is published
  - Simplified implementation (no append-only or fault tolerance mechanisms)
- Distribution of trust
  - Shared decryption key (3 decryptors, no threshold)
  - Two mix networks (each with 3 mixers, no proof yet)
- Extended voter privacy
  - Secrecy: mixing the votes
  - Anonymity: mixing the public signature keys
- Transparency (source code and documentation)
Design and Specification

Rolf Haenni & Oliver Spycher

Secure Internet Voting on Limited Devices with Anonymized Public Keys

EVT/WOTE'11
Design and Specification

Rolf Haenni & Oliver Spycher
Secure Internet Voting on Limited Devices with Anonymized Public Keys
EVT/WOTE'11

Rolf Haenni
UniVote System Specification
v.0.5.2
1.3.7. Mixing and Tallying

a) Mixing the Encryptions

Let $E_0 = \{E_1, \ldots, E_N\}$, $N \leq n$, be the (ordered) set of encrypted votes in $B$. Repeat the following steps for each $M_k \in M$ (in ascending order for $1 \leq k \leq m$):

1. Shuffle the set encrypted votes $E_{k-1}$ into $E_k$:
   
   a) Choose $\bar{r}_k = (r_{1k}, \ldots, r_{Nk}) \in_R \mathbb{Z}_q^N$ uniformly at random and compute $E'_i = ReEnc_y(E_i, r_{ik})$ for every $E_i \in E_{k-1}$.

   b) Choose permutation $\tau_k : [1, N] \rightarrow [1, N]$ uniformly at random.

   c) Let $E_k = \{E'_{\tau_k(i)} : 1 \leq i \leq N\} = \text{Shuffle}_{\tau_k}(E_{k-1}, \bar{r}_k)$ be the new (ordered) set of encrypted votes shuffled according to $\tau_k$.

2. Generate $\pi_{\tau_k} = \text{NIZKP}\{(\tau_k, \bar{r}_k) : E_k = \text{Shuffle}_{\tau_k}(E_{k-1}, \bar{r}_k)\}$ using Wikström’s proof of a shuffle (see Section 1.4.7 for details).

3. Generate signature $S_{E_k} = \text{Sign}_{sk_k}(id||E_k||\pi_{\tau_k})$.

4. Publish $(M_k, id, E_k, \pi_{\tau_k}, S_{E_k})$ on $EB$. 
Rolf Haenni & Oliver Spycher
Secure Internet Voting on Limited Devices with Anonymized Public Keys
EVT/WOTE'11

Rolf Haenni
UniVote System Specification
v.0.5.2

UniVote
v.1.7

UniCrypt (Cryptographic Library)
Tools and Components
UniCrypt

- Java library for advanced cryptographic tasks
  - ElGamal encryptions
  - Commitments
  - Secret sharing
  - Re-encryption mixnets
  - Zero-knowledge proofs
  - Elliptic curves
  - Random oracles
  - Common reference strings

- Design goal: Clean and easy-to-use programming interfaces

- Version 2.0 to be released soon (summer 2014)

- Open-source project on GitHub:
  https://github.com/bfh-evg/unicrypt
Public Election Board
Public Election Board

Voter → UniVote v.1.7 → Result

Public Election Board

Berne University of Applied Sciences | Berner Fachhochschule | Haute cole specialise bernoise
Public Election Board

Voter → UniVote v.1.7 → Public Election Board → VoteVerifier v.1.0 → Result
Voter Verifier

- Student project (bachelor thesis 2013)
- Developed independently from specification
  - Disjoint code base
  - No help from UniVote source code
- Individual verification:
  - Reads encrypted vote from QR-code
  - Checks if encrypted vote has reached the election board
  - Displays vote intention on trustworthy device
- Universal verification:
  - Reads election data from public election board
  - Checks consistency of election data
  - Total of 61 checks: parameters, signatures, crypto-proofs
  - Re-computes final election result
## Election Results

### FVV
- 1.1 Cornelia Vontobel: 132
- 1.2 Saskia Keller: 108

### IG Oerlikon
- 2.1 Ivan Marijanovic: 852
- 2.2 Roberto Ramphos: 739
- 2.3 Muriel Ehrbar: 775
- 2.4 Nadja Busch: 756
- 2.5 Nina Egger: 776
- 2.6 Tristan Jennings: 727
- 2.7 Louis Binswanger: 710
Vote Verifier

Voter

UniVote
v.1.7

Public Election Board

VoteVerifier
v.1.0

Result
VoteSimulator

- Answer to question: “Who checks the VoteVerifier?”
- Student project at HSR (work in progress)
- Developed independently from specification
  - Disjoint code base
  - No help from UniVote source code
  - No help from VoteVerifier source code
- Writes data for arbitrary-sized elections to election board
  - Good case: consistent data only
  - Bad case: inconsistent data from simulated attacks
Conclusion and Future Work
Conclusion

- For academics, it is very instructive . . .
  - to develop a real-world election system
  - to run real elections

- Student board elections are a great testbed

- Very positive feedback . . .
  - from voters
  - from research community

- Major problems
  - Small budget
  - Restricted manpower
  - Time management
  - Browser incompatibilities
  - Software maintenance (students disappear after graduating)
UniVote 2.0

- UniVote 2.0 = Complete redesign of UniVote 1.7
  - Independent append-only public election board (UniBoard)
  - Improved underlying cryptographic library (UniCrypt)
  - Extended independent registration service (UniCert) for Google+, Facebook, Twitter, etc.
  - GUI support for multiple election types
  - Improved election administration tools
  - Comprehensive documentation

- Enlarged project team
  - 2 PhD students
  - 1 full time assistant

- Lack of project funding
UniVote 2.0

UniVote 2.0 = Complete redesign of UniVote 1.7

- Independent append-only public election board (UniBoard)
- Improved underlying cryptographic library (UniCrypt)
- Extended independent registration service (UniCert) for Google+, Facebook, Twitter, etc.
- GUI support for multiple election types
- Improved election administration tools
- Comprehensive documentation

Enlarged project team

- 2 PhD students
- 1 full time assistant

Lack of project funding