Internet voting by CHvote

@SwissCyberStorm

18th october 2017
Short Bio

Thomas Hofer / @thhofer / thomas.hofer@etat.ge.ch

- EPFL MSc in IT
- Master thesis @ CERN Computer Security Team
- IT / Java consultant

Now
- Java DEV & AppSec
- Internet voting cryptography @ State of Geneva

Outside from work
- OWASP-Geneva co-chapter leader
- Married, 2 kids
Outline

CHvote

Control components: in theory

Control components: in practice

Questions
Outline

- CHvote
- Control components: in theory
- Control components: in practice
- Questions
The past of CHvote

First generation E-Voting system

- 2001: start of project
- 2003: first use

• Partners
The present of CHvote

Individual verifiability & major appearance overhaul
The future of CHvote
End-to-end verifiable internet voting protocol

• New academic partnerships
  → Berner Fachhochschule
  → INRIA / Bristol
  → ITU Copenhagen
  → ...

• New cryptographic protocol
  → End-to-end encryption
  → Universal Verifiability
  → Control Components

• Currently in development, ETA: 2019
Outline

- CHvote
- Control components: in theory
- Control components: in practice
- Questions
Federal requirements
New Ordinance on Electronic Voting

• Published in 2013, enacted 2014
  → Collaborative work between lawmakers, academia and operating staff

• Compliance levels
  → The higher the compliance, the more voters allowed

• Reference
Federal requirements

Control Components

The trustworthy part of the system includes either one or a small number of groups of independent components secured by special measures (control components). Their use must also make any abuse recognisable if per group only one of the control components works correctly and in particular is not manipulated unnoticed. – VEleS, art. 5, par. 6

• i.e. trust splitting in "anytrust" mode: only 1-of-k needs to be honest, all k need to be available

• Control components must use different OS
Outline

CHvote

Control components: in theory

Control components: in practice

Questions
Security role of control components

A brief overview

- Homomorphic encryption (El Gamal)
- Public / private credentials per voter
  → Each Control Component holds a share
- 1 key pair per Control Component
  → "system" public key is the product of public keys
- 1 ballot-box shuffle per Control Component
Voter authentication

In the context of trust splitting

• Private credentials:
  → Signed and encrypted for printing authority
  → Used for signing the ballot

• Public credentials:
  → Shared and combined
  → Each Control Component must verify credentials
Ballot encryption

How?

• Voting client uses "system" key to encrypt ballot
  → "system" key is product of each CC's public key

• All Control Components are required for decryption

• (partial) Decryption by CC is not allowed until:
  → Ballot box has been shuffled by all CCs
  → Each shuffle has been verified to be valid
Oblivious Transfer

What does it mean and why is it useful?

• In short
  → Server knows n secret messages
  → Client allowed to retrieve k secret messages
  → Server cannot know which messages the client asked for
  → Enables individual verifiability combined with end-to-end encryption

• In detail
  → Cast-as-Intended Verification in Electronic Elections Based on Oblivious Transfer
Ballot-box shuffling

Why is it complicated and how does it work

• Re-encrypting mix-net
  → Each CC re-encrypts each ballot and shuffles them

• Shuffled → simple pre-image proofs would not work
• Re-encrypted → ciphertexts are not equal

• Need for a specific proof that the cryptographic shuffle is valid
Further reading

And references

• Published protocol specification
  → https://eprint.iacr.org/2017/325

• Published PoC code
  → https://github.com/republique-et-canton-de-geneve/chvote-protocol-poc

• Federal requirements
Thank you!

Thomas Hofer

thomas.hofer@etat.ge.ch

@thhofer