Bern University
of Applied Sciences

Private Internet Voting on Untrusted Voting Devices

Rolf Haenni, Reto E. Koenig, Philipp Locher (BFH)

Cyber Alp Retreat, Flüeli-Ranft, Switzerland
June 17–18, 2024

Outline

- Introduction
- Voting Procedure
- Cryptographic Protocol
- Security and Usability Properties
- Conclusion

Outline

E-Voting in Switzerland

Source: https://www.zdnet.com/article/swiss-government-invites-hackers-to-pen-test-its-e-voting-system

Bericht
des Bundesrates zu Vote électronique

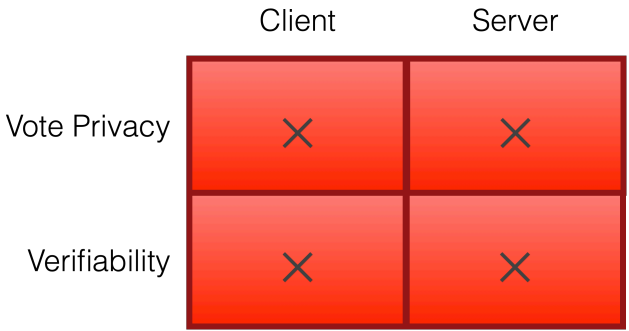
Auswertung der Einführung von Vote électronique (2006–2012)
und Grundlagen zur Weiterentwicklung

vom 14. Juni 2013

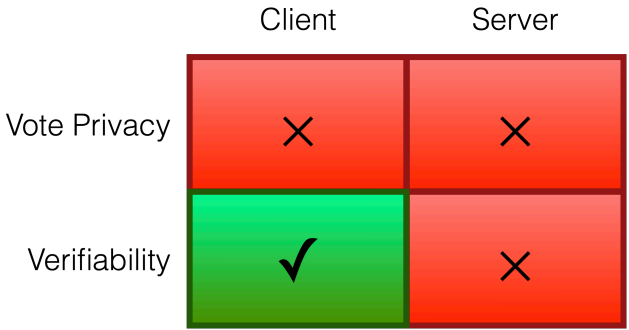
Neue Sicherheitsanforderungen (Ziff. 12):
Im Zentrum der Sicherheitsanforderungen steht die Verifizierbarkeit

(Verifiability is at the centre of the security requirements)

Systems of 1st Generation: 2004–2014

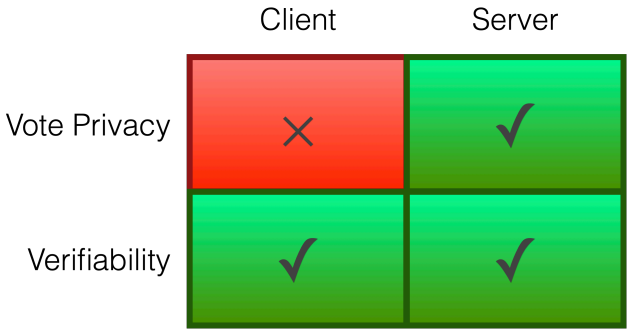


Systems of 2nd Generation: 2014–2018



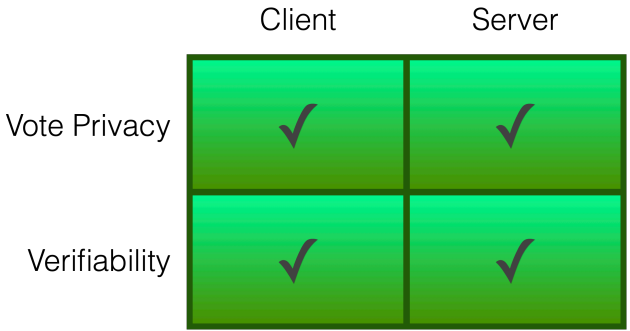
Die Schweizerische Bundeskanzlei (BK), *Verordnung der Bundeskanzlei über die elektronische Stimmabgabe* (VEleS), December 2013

Systems of 3rd Generation: since 2018



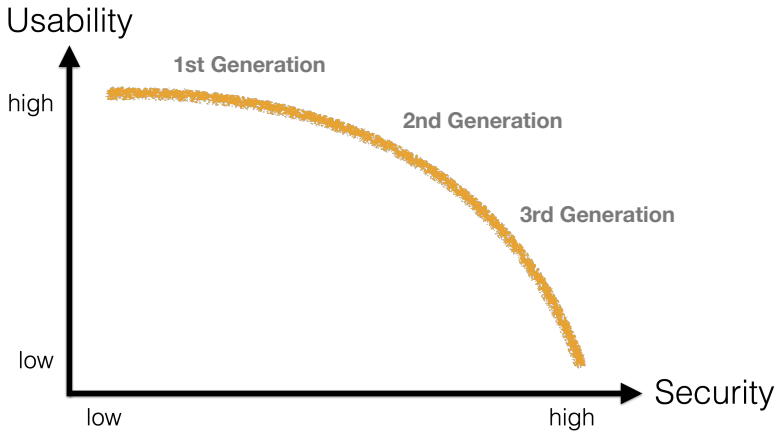
Die Schweizerische Bundeskanzlei (BK), *Verordnung der Bundeskanzlei über die elektronische Stimmabgabe* (VEleS), July 2018

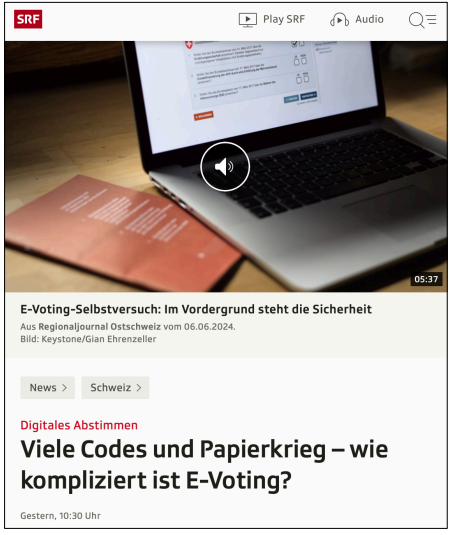
Systems of 4th Generation: this talk!



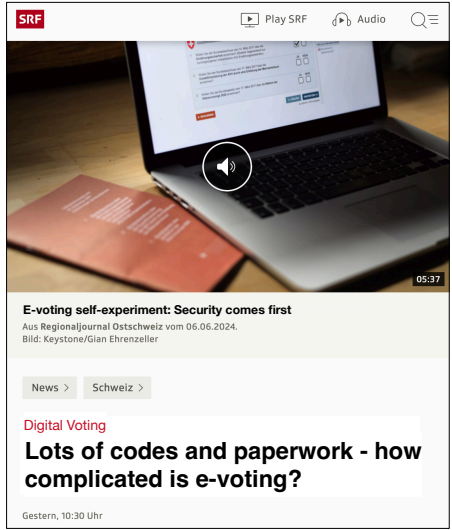
R. Haenni, R. E. Koenig, and P. Locher. *Private internet voting on untrusted voting devices*. Voting'23, FC 2023 International Workshops, LNCS 13953, Brac, Croatia, 2023

Security vs. Usability





https://www.srf.ch/news/schweiz/digitales-abstimmen-viele-codes-und-papierkrieg-wie-kompliziert-ist-e-voting [June 16, 2024]



https://www.srf.ch/news/schweiz/digitales-abstimmen-viele-codes-und-papierkrieg-wie-kompliziert-ist-e-voting [June 16, 2024]

QR-Codes are Everywhere

Source: https://www.freepik.com/premium-vector/flat-boarding-pass-with-qr-code-red-shapes_1005491.htm

QR-Codes are Everywhere

Source: https://www.qrcode-tiger.com/6-interesting-use-cases-of-qr-codes-in-2019-and-how-it-can-benefit-your-business

QR-Codes are Everywhere

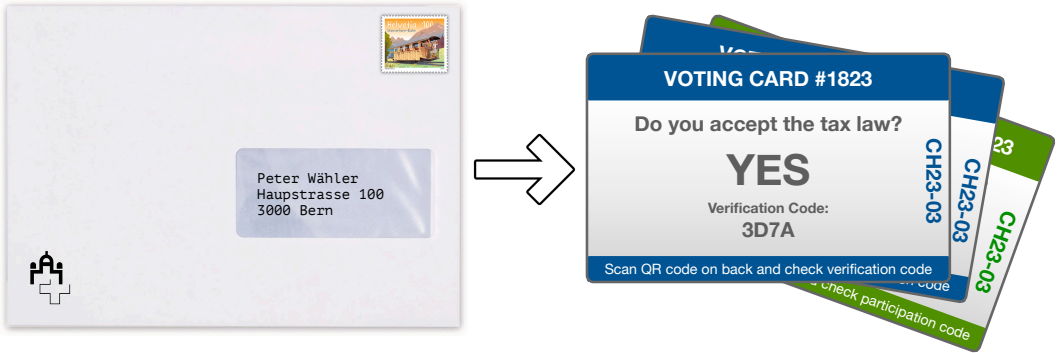
Source: https://krvportal.ch/rechnungen/qr-rechnung/

Outline

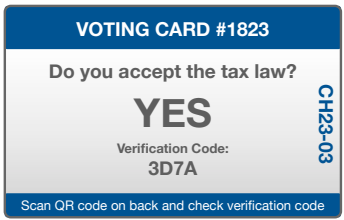
- Introduction
- Voting Procedure
- Cryptographic Protocol
- Security and Usability Properties
- Conclusion

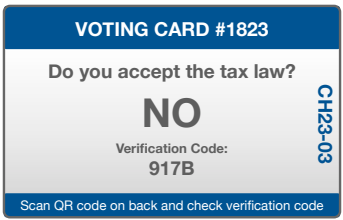
Step 1: Invitation by Postal Mail

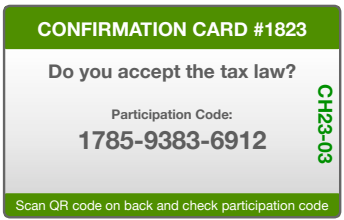
Step 1: Invitation by Postal Mail



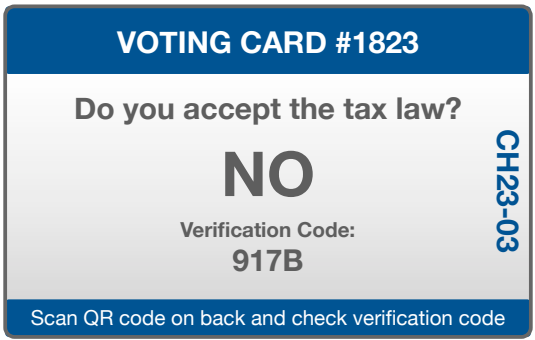
Step 1: Invitation by Postal Mail







Step 2: Vote Casting



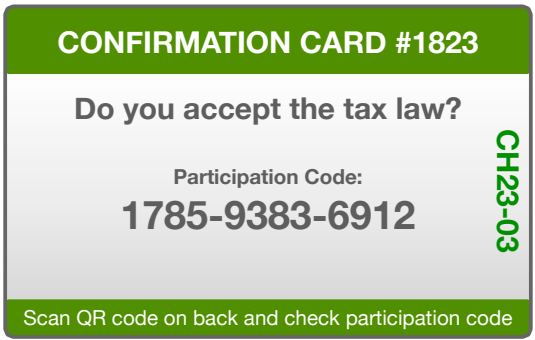
Step 2: Vote Casting

Step 2: Vote Casting

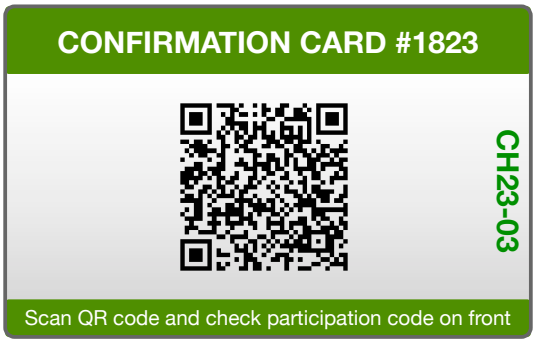
Step 2: Vote Casting

Step 2: Vote Casting

Step 3: Confirmation



Step 3: Confirmation



Step 3: Confirmation

Step 3: Confirmation

Step 3: Confirmation

Usability

- Voter's can use their regular phone, tables, or notebook computer
- Multiple voters can use the same device
- No additional voting software needs to be installed
- No URL needs to be entered
- No passwords or keys need to be entered
- For a small number of selections m (e.g. in a referendum), vote casting can be done in a very short amount of time

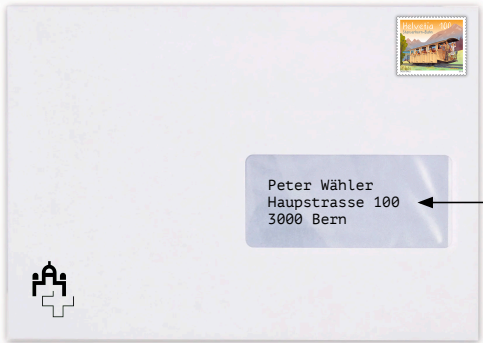
Outline

Preliminaries: Election Parameters

- N = number of voters
- n = number of voting options (candidates)
- m = maximum number of selection

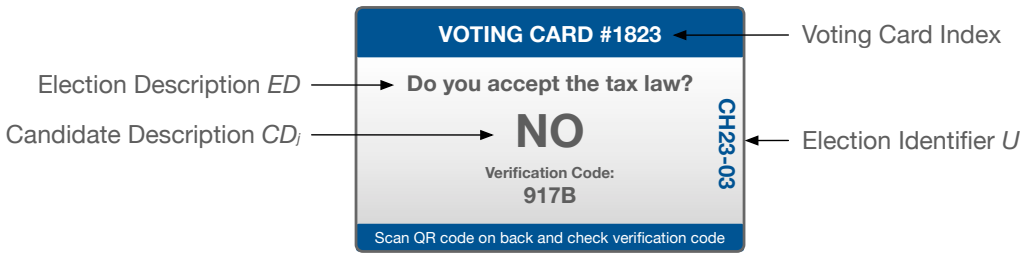
- U = unique election identifier
- ED = election description
- VD_i = voter description $i \in [1, N]$
- CD_j = candidate description $j \in [1, n]$

Preliminaries: Election Parameters

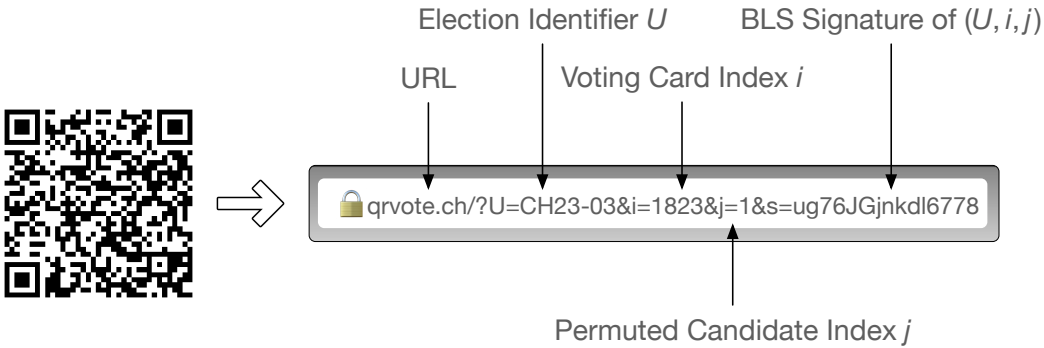


Voter Description VD_i

Preliminaries: Election Parameters



Preliminaries: Election Parameters



Preliminaries: Cryptographic Parameters

- $\mathbb{G}_q$ = prime-order subgroup of integers modulo $p = 2q + 1$
- Generator $g \in \mathbb{G}_q \setminus \{1\}$
- Vote encoding $\Gamma : [1, n] \to \mathbb{G}_q$ and decoding $\Gamma^{-1} : \mathbb{G}_q \to [1, n]$

- Bilinear map $e : G_1 \times G_2 \to G_T$, for example BLS12-381
- Hash function $hash : \{0, 1\}^* \to G_1$
- Generator $g_2 \in G_2 \setminus \{1\}$
- Note that the BLS signature size in BLS12-381 is 382 bits (48 bytes)

Preliminaries: Cryptographic Keys

- s = number of election authorities

- dk_k = private ElGamal decryption key of authority $k \in [1, s]$
- ek = jointly generated ElGamal encryption public key

- sk_k = private BLS signature generation key of authority $k \in [1, s]$
- vk = jointly generated BLS signature verification public key

Preparation: Mixing the Encrypted Votes

- For all $j \in [1, n]$, let $e_j = Enc_{ek}(\Gamma(j), 0) = (1, \Gamma(j))$ be the trivial encryption of j
- For voting card $i \in [1, N]$, apply a verifiable mixnet to $\tilde{\mathbf{e}}_{i,0} = (e_1, \ldots, e_n)$:

$$(e_1, \ldots, e_n) \xrightarrow[\psi_{i,1}]{\text{Shuffle 1}} \tilde{\mathbf{e}}_{i,1} \xrightarrow[\psi_{i,2}]{\text{Shuffle 2}} \cdots \xrightarrow[\psi_{i,s},]{\text{Shuffle } s} \tilde{\mathbf{e}}_{i,s},$$

and let $\tilde{\mathbf{e}}_{i,s} = (\tilde{e}_{i,1}, \ldots, \tilde{e}_{i,n})$ be its output (together with NIZKPs for all steps)

- The shuffle permutations ψ_{ik} are sent to the trusted printer (by opening the commitments from the NIZKP)
- The printer combines $\psi_i = \psi_{i,s} \circ \cdots \circ \psi_{i,1}$ and therefore knows the plaintext candidates of each encryption $\tilde{e}_{ij}$

Preparation: Preparing the Voting Cards

- Authority $k \in [1, s]$ generates BLS signatures $s_{ijk} = sign_{sk_k}(U, i, j)$ and random verification codes vc_{ijk} for all voters i and all candidates j
- The resulting matrices $(s_{ijk})_{N \times n}$ and $(vc_{ijk})_{N \times n}$ are sent to the trusted printer

- The printer combines $s_{ij} = \prod_{k=1}^{s} s_{ijk}$ and $vc_{ij} = hash(vc_{ij1}, \ldots, vc_{ijk})$
- The printer computes $QR_{ij} = QREncode(U, i, j, s_{ij})$
- The printer assign QR_{ij} and vc_{ij} to the candidate description $CD_{\psi_i^{-1}(j)}$
- The printed voting cards are sent to the voters

- Similar for confirmation cards

Vote Casting and Confirmation

- The voter sends QR_{ij} to the authorities
- The vote is accepted if s_{ij} is a valid signature for (U, i, j)
- Shares of the verification code vc_{ijk} are sent to the voter
- The voting client displays $vc_{ij} = hash(vc_{ij1}, \ldots, vc_{ijk})$ to the voter

- Similar procedure for vote confirmation

Tallying

- Apply a verifiable mixnet to the accepted encrypted votes
- Perform a distributed decryption using the shares dk_k of the private decryption key
- Decode the plaintext votes using Γ^{-1}
- Publish the election result

Outline

Security: Vote Privacy

- Assumption 1: The printer is honest
- Assumption 2: The election authorities are honest as a group (at least is honest)

- The combined permutation ψ_i of the pre-election mixnet is only known to the trusted printer
- Therefore, no adversary can link the submitted permuted candidate index to the candidate selected by the voter
- In particular, the voting device learns nothing about the voter's intention

- The post-election mixnet unlinks the decrypted votes from the submitted votes

Security: Vote Integrity

- Assumption 1: The printer is honest
- Assumption 2: The election authorities are honest as a group (at least is honest)

- Only ballots $b_{ij} = (U, i, j, s_{ij})$ with a valid signature make it into the final tally
- Valid signatures are only known to the trusted printer and the voters
- No single party can generate valid signatures

- The untrusted voting device could try to change or block the submitted vote, but this would ne noticed by the voter

Security: Verifiability

- Preparation: input/output/NIZKP of mixnet are public
- Vote Casting: submitted ballots are public (incl. BLS signatures)
- Tallying: input/output/NIZKP of mixnet, decryption proofs, election result are public, decoding function is deterministic

Outline

Conclusion

Achievements

- Approach inspired by Chaum's original code voting scheme
- BLS signatures and mixnets help to weaken trust assumptions
- Popularity of QR codes helps to offer good usability
- Proof-of-concept implementation done by student (2023) ¶

Open problems

- Usability in complex elections
- Security proofs

Questions?