



Berner Fachhochschule  
Haute école spécialisée bernoise  
Bern University of Applied Sciences



# Öffentlicher Intrusionstest der Post: Was wir lernen können

Prof. Dr. Eric Dubuis

Vortrag 19. Rechtsinformatikseminar 13. und 14. Mai 2019, Magglingen

► Technik und Informatik, RISIS

# Eric Dubuis

- ▶ Professor für Informatik
- ▶ Departement Technik und Informatik
- ▶ Leiter des Research Institute for Security in the Information Society (RISIS) / BFH-Zentrum «Digital Society»
- ▶ Mitbegründer des Swiss E-Voting Competence Center (nicht mehr aktiv)
- ▶ Seit 2008 Forschung zu elektronischen Wahlen übers Internet (im Team!)

# Agenda

- ▶ Einleitung: Der öffentliche Intrusionstest der Post
- ▶ Exkurs: Was macht E-Voting so schwierig?
- ▶ Zurück zum Wesentlichen: Was ist passiert?
- ▶ Die Lehren: Wie könnte das vermieden werden?

# Öffentlicher Intrusionstest der Post

### Aktuelles zu E-Voting

#### Post lässt ihr Abstimmungssystem durch Hacker auf Herz und Nieren testen

Die Schweizerische Post setzt vom 25. Februar bis 24. März 2019 ihr E-Voting System einem öffentlichen Hackertest aus. Damit erfüllt sie die Vorgaben von Bund und Kantonen. Mit dem sogenannten «öffentlichen Intrusionstest» können angemeldete IT-Spezialisten das System auf Herz und Nieren prüfen und bei einem fiktiven Urnengang das Resultat zu manipulieren versuchen. Die Post wird die Ergebnisse des Hackertests in die Weiterentwicklung ihres E-Voting-Systems aufnehmen. Am 7. Februar veröffentlichte die Post zudem den Quellcode ihres Systems. Unabhängige Experten können auch diesen kritisch prüfen und sich eingehend auf den öffentlichen Intrusionstest vorbereiten.

<https://www.post.ch/de/geschaefftlich/themen-a-z/branchenloesungen/e-voting-loesung-der-post>

Stiess bei der E-Voting-Community auf positives Echo!

# Was macht E-Voting so schwierig?

# Was soll ein Wahlsystem garantieren?

## «Demokratie»-Regeln:

- ▶ Nur Stimmen von Stimmberechtigten fließen ins Resultat ein («**Berechtigung**»)
- ▶ Eine stimmberechtigte Person  $\leftrightarrow$  eine Stimme («**Eine-Stimme-Eigenschaft**»)
- ▶ Das Resultat ist erst nach Urnenschluss bekannt («**Fairness**»)

## Regeln zum Schutz der Privatsphäre:

- ▶ Die Stimme ist geheim («**Stimmgeheimnis**»)
- ▶ Abstimmende Person kann nicht beweisen, wie sie gestimmt hat («**keine Quittung**»)

## Verifizierbarkeit:

- ▶ Wurde meine Stimme richtig gezählt? («**individuelle Verifizierbarkeit**»)
- ▶ Wurden alle gültigen Stimmen gezählt? («**universelle Verifizierbarkeit**»)

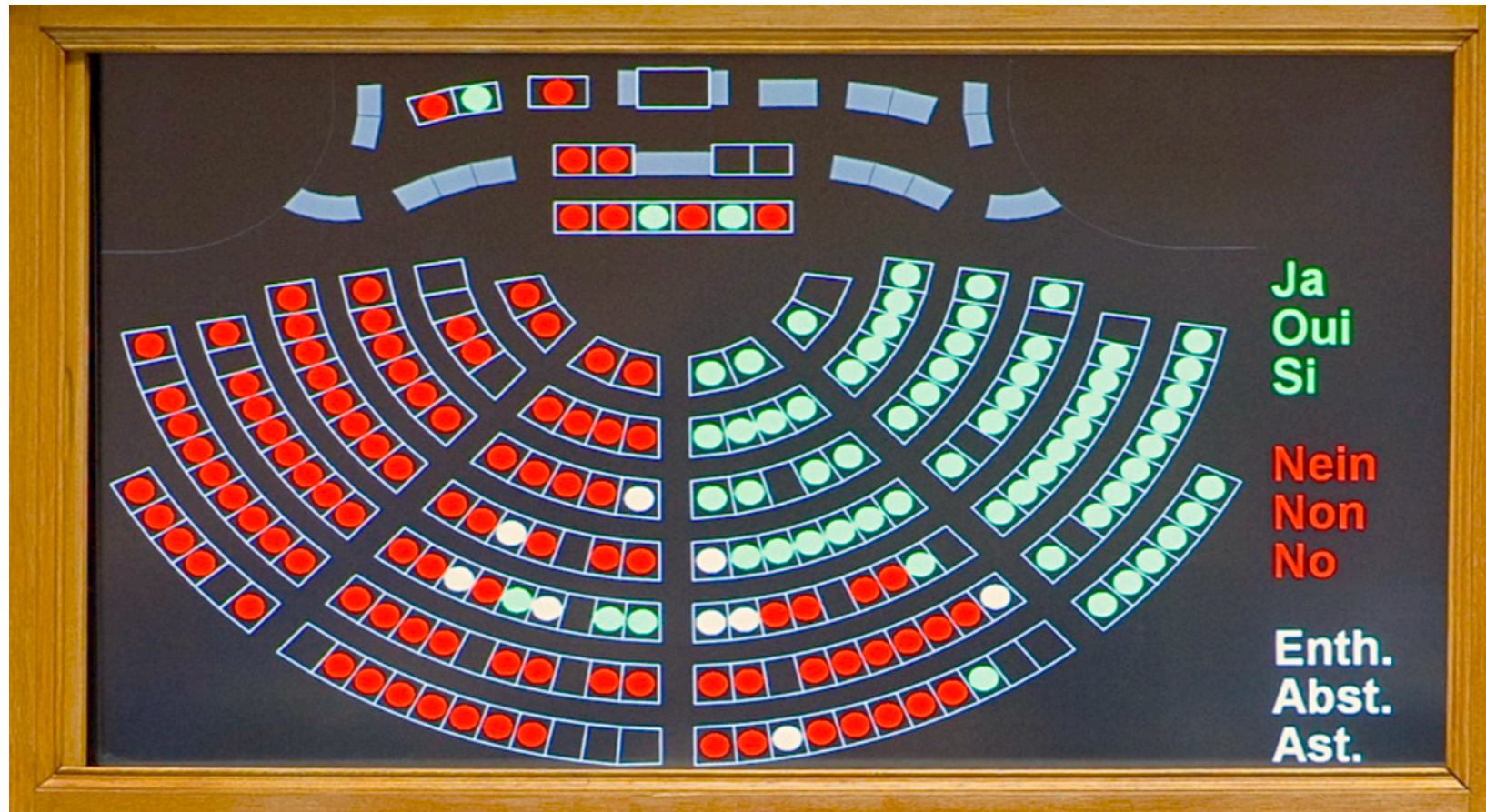
# Herausforderungen

Entwicklung eines E-Voting-Systems mit teilweise widersprüchlichen Anforderungen:

- ▶ Wahlberechtigung  $\leftrightarrow$  Wahlgeheimnis
- ▶ Verifizierbarkeit  $\leftrightarrow$  Wahlgeheimnis



# Universelle Verifizierung einfach erklärt



Quelle: srf.ch

Was ist passiert?

Abonnemente

Neue Zürcher Zeitung

**BZ** BERNER ZEITUNG

Region Sport Schweiz Wirtschaft Ausland Panorama

# Experten finden Lücke im E-Voting-System der Post

Eine Sicherheitslücke hätte ermöglicht, Abstimmungsergebnisse zu manipulieren, ohne dass dies entdeckt worden wäre.

2019-03-12 13:03

# Fehler in *Commitments*

$$c = G_1^{m_1} \dots G_n^{m_n} H^r$$

$G_1, \dots, G_n, H$ : müssen paarweise unabhängig sein

# Mögliche Auswirkung der fehlerhaften *Commitments*

- 1) Das E-Voting-System könnte Stimmen manipulieren, so dass am Ende das gewünschte Ergebnis vorläge
- 2) Das E-Voting-System könnte Stimmen markieren, so dass nach erfolgter Entschlüsselung dank der Marken das Stimmgeheimnis gebrochen würde

In beiden Fällen würde die durchgeführte universelle Verifikation die Manipulation nicht entdecken!

# Behebung des Fehlers

- ▶ NIST Standard FIPS PUB 186-4 [2,Appendix A.2.3]  
«Digital Signature Standard»

<https://csrc.nist.gov/publications/detail/fips/186/4/final>

# Fehler in kryptografischen Beweisen

► Verschlüsselung:

$$(C_0, C_1) = (g^r, m(pk)^r)$$

$m$ : Stimme

$pk = g^x$  mit  $x$  als geheimen Schlüssel

► Beweis für eine Entschlüsselung:

1. Pick a random  $a$ .

2. set  $B_0 = g^a$  and  $B_1 = C_0^a$ .

3. **Compute**  $c = \mathbf{H}(pk, C_1, B_0, B_1)$ ,

4. Compute  $z = a + cx$ .

► Beweis:

$$(c, z)$$

# Weshalb manipuliert werden könnte

► Beweis:

$$(c, z)$$

► Verifikation:

$$B_0 = g^z(pk)^{-c}$$

$$B_1 = C_0^z(C_1)^{-c}$$

$$c \stackrel{?}{=} H(pk, C_1, B_0, B_1)$$



# Mögliche Auswirkung dieser Manipulation

- 1) Das E-Voting-System könnte Stimmen manipulieren, so dass sie am Ende ungültig wären und somit nicht in die Ermittlung des Ergebnisses einfließen könnten
- 2) Schadsoftware im Computer des Stimmenden könnten anstelle korrekter Stimmen Unsinn verschlüsseln

In beiden Fällen würde die durchgeführte universelle Verifikation die Manipulation nicht entdecken!

# Behebung des Fehlers

$(c, z)$

$$B_0 = g^z (pk)^{-c}$$

$$B_1 = C_0^z (C_1)^{-c}$$

$$c \stackrel{?}{=} H(pk, C_1, B_0, B_1)$$

- ▶ Fehler:  $g$  und  $C_0$  müssen «eingepackt» werden:

$$\mathbf{c} = \mathbf{H}(\mathbf{pk}, \mathbf{C}_1, \mathbf{B}_0, \mathbf{B}_1)$$

Wie könnte dies vermieden werden?

# Konsequente Umsetzung von Vertrauen bildenden Massnahmen (I)

- ▶ Spezifikation der Systeme offenlegen → Verifikations-Software!
- ▶ Alle weiteren Dokumente offenlegen wie
  - ▶ Sicherheitskonzept
  - ▶ Sicherheitsbeweise
- ▶ Zertifizierungsprozess überdenken
- ▶ Vertrauensannahmen hinterfragen

# Übersicht der Vertrauensannahmen bei der Post

sVote's system component	Chancellery's system component	Trust assumption
Voters	Voters	<b>significant proportion</b> of voters are <b>non-trustworthy</b>
Voting Client	User platform	<b>untrustworthy</b> for individual and complete verifiability <b>trustworthy</b> for privacy
Voting Card	Trusted technical aids for voters	<b>trustworthy</b>
Voting Server	System (server-side)	<b>untrustworthy</b>
Print office	Print office	<b>trustworthy</b>
CCM CCR	Control Components	<b>trustworthy only as the whole</b>
Auditors	Auditors	at least <b>one is trustworthy</b>
Verifier	Auditor's technical aid	at least one honest auditor has a <b>trustworthy</b> aid

<https://www.post.ch/-/media/post/evoting/dokumente/complete-verifiability-security-proof-report.pdf>

# Konsequente Umsetzung von Vertrauen bildenden Massnahmen (II)

- ▶ Spezifikation der Systeme offenlegen → Verifikations-Software!
- ▶ Alle weiteren Dokumente offenlegen wie
  - ▶ Sicherheitskonzept
  - ▶ Sicherheitsbeweise
- ▶ Zertifizierungsprozess überdenken
- ▶ Vertrauensannahmen hinterfragen
- ▶ Quellcode offenlegen → Umsetzungsfehler!

# Vielen Dank / Fragen?

Prof. Dr. Eric Dubuis  
RISIS / E-Voting-Gruppe  
Zentrum Digital Society

