Bern University
of Applied Sciences

# CHVote Voting Protocol

*Rolf Haenni*

April 19th, 2018

# Outline

- Introduction
- Cast-as-Intended Verification
- CHVote Voting Protocol
- Conclusion

# Introduction

# Direct Democracy in Switzerland

- Up to four election days per year
  - Elections
  - Mandatory referendums
  - Optional referendums ($>$50k signatures)
  - Popular initiatives ($>$100k signatures)
- Three different political levels
  - Federal
  - Cantonal
  - Municipal
- Up to 10 different election topics per election day

# E-Voting Tradition in Switzerland

- Classical voting channels
  - Polling station
  - Landsgemeinde
  - Postal voting (since 1994, today approx. 90%)
- Non-verifiable "blackbox" e-voting systems (1st generation)
  - Canton of Geneva (since 2003)
  - Canton of Zürich (Unisys, 2004–2015)
  - Canton of Neuchâtel (Scytl, 2005–2015)
- Collaborations with 10 other cantons (since 2009)

*The introduction of verifiability is central to the new security requirements.*

3rd Vote Electronique Report
Swiss Federal Council, 2013

# Legal Ordinance on Electronic Voting

- Effective since December 2013
- Enhanced security requirements
  - End-to-end encryption
  - Individual verifiability (cast-as-intended, recorded-as-cast)
  - Universal verifiability
  - Distribution of trust (shared decryption key, mix-net)
- Two-step expansion
  - Current systems: max. 10% of federal electorate
  - Step 1: max. 30% of federal electorate
  - Step 2: 100% electorate

# Cast-as-Intended Verification

# Cast-as-Intended Verification

- Prior to an election, a code sheet with different verification codes for each voting option is generated for every voter
- Verification codes are different on every code sheet
- Code sheets are sent to voters by postal mail

**Code Sheet Nr.291**

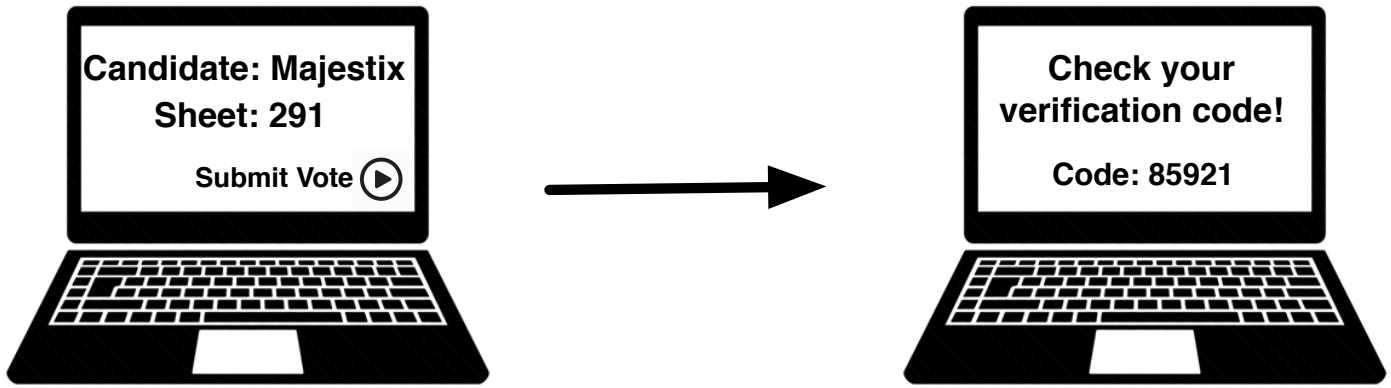
| Candidates | Codes |
|---|---|
| Asterix | 74494 |
| Obelix | 84443 |
| Idefix | 91123 |
| Miraculix | 63382 |
| Majestix | 85921 |
| Verleihnix | 79174 |

**Code Sheet Nr.321**

| Candidates | Codes |
|---|---|
| Asterix | 21344 |
| Obelix | 29173 |
| Idefix | 91123 |
| Miraculix | 72282 |
| Majestix | 18194 |
| Verleihnix | 53382 |

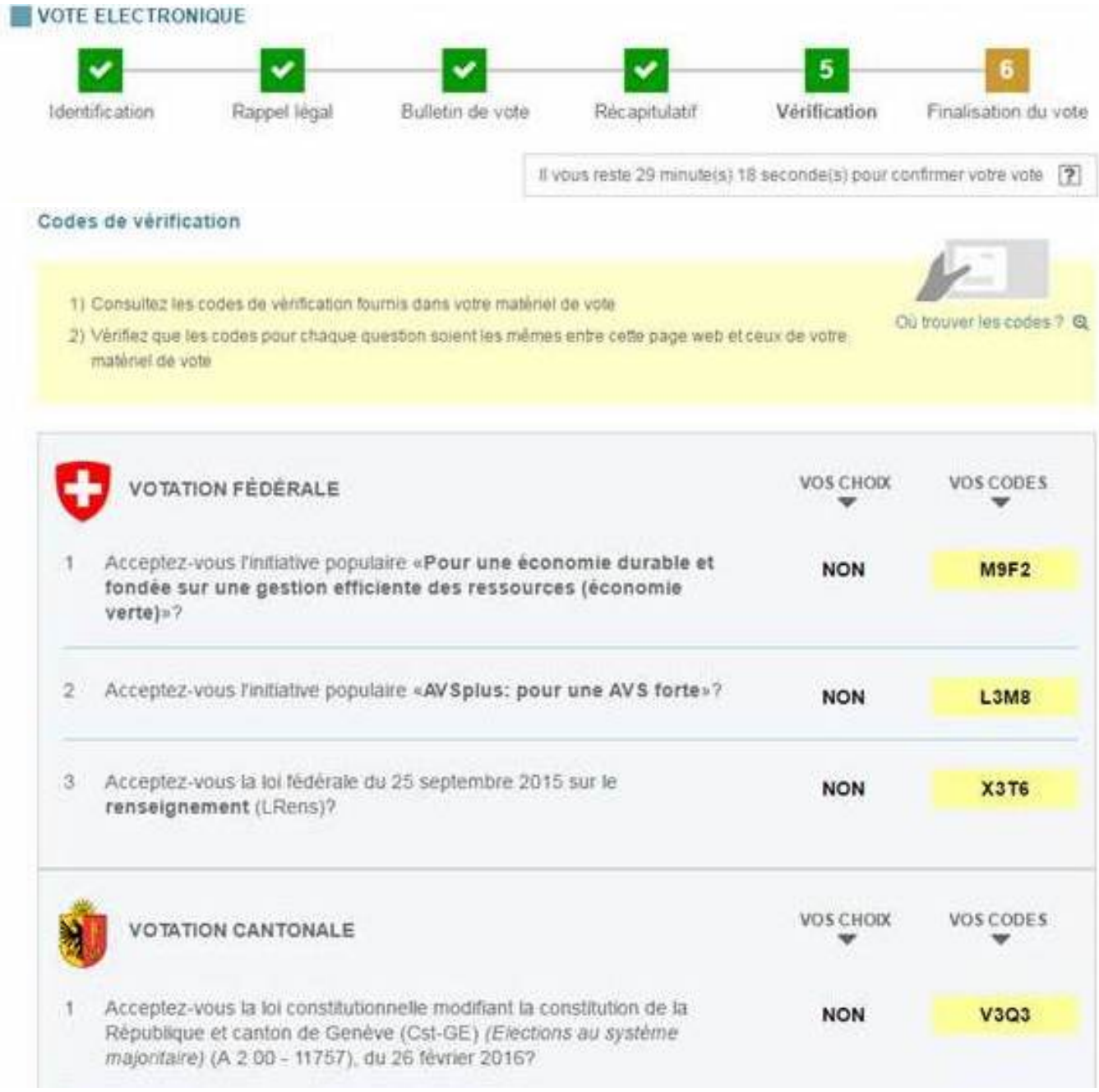
# Cast-as-Intended Verification

- After submitting a vote, corresponding verification codes are displayed




- Matching codes imply that the vote has been cast as intended
- Otherwise, voters are instructed to vote by postal mail

| Liste de codes pour la carte n° 5874-8863-1400-8743 | | | |
|---|---|---|---|
| **Votation fédérale** | | | |
| Question 1<br>Acceptez-vous l'arrêté fédéral du 20 juin 2013 portant règlement du financement et de l'aménagement de l'infrastructure ferroviaire (Contre-projet direct à l'initiative populaire "Pour les transports publics", qui a été retirée) ? | Oui<br>A2B4 | Non<br>J5B9 | Blanc<br>Z8H5 |
| Question 2<br>Acceptez-vous l'initiative populaire "Financer l'avortement est une affaire privée - Alléger l'assurance-maladie en radiant les coûts de l'interruption de grossesse de l'assurance de base" ? | Oui<br>P8H3 | Non<br>X2A7 | Blanc<br>Q3L7 |
| **Votation cantonale** | | | |
| Question 1<br>Acceptez-vous l'initiative 143 «Pour une véritable politique d'accueil de la Petite enfance» ? | Oui<br>U6T4 | Non<br>P3D6 | Blanc<br>S6C2 |
| Question 2<br>Acceptez-vous la loi constitutionnelle modifiant la constitution de la République et canton de Genève (Contreprojet à l'IN 143) (A 2 00 – 10895), du 15 décembre 2011 ? | Oui<br>N4F2 | Non<br>M2A3 | Blanc<br>Q9L5 |
| Question 3<br>**Question subsidiaire**: Si l'initiative (IN 143 «Pour une véritable politique d'accueil de la Petite enfance») et le contreprojet sont acceptés, lequel des deux a-t-il votre préférence ? Initiative 143 ? Contreprojet ? | IN<br>K9W9 | CP<br>T3S6 | Blanc<br>Y2V4 |


VOTE ELECTRONIQUE
Identification
Rappel légal
Bulletin de vote
Récapitulatif
5
Vérification
6
Finalisation du vote
Il vous reste 29 minute(s) 18 seconde(s) pour confirmer votre vote
Codes de vérification
1) Consultez les codes de vérification fournis dans votre matériel de vote
2) Vérifiez que les codes pour chaque question soient les mêmes entre cette page web et ceux de votre matériel de vote
Où trouver les codes ?
VOTATION FÉDÉRALE
VOS CHOIX
VOS CODES
1 Acceptez-vous l'initiative populaire «Pour une économie durable et fondée sur une gestion efficiente des ressources (économie verte)»? NON M9F2
2 Acceptez-vous l'initiative populaire «AVSplus: pour une AVS forte»? NON L3M8
3 Acceptez-vous la loi fédérale du 25 septembre 2015 sur le renseignement (LRens)? NON X3T6
VOTATION CANTONALE
VOS CHOIX
VOS CODES
1 Acceptez-vous la loi constitutionnelle modifiant la constitution de la République et canton de Genève (Cst-GE) (Elections au système majoritaire) (A 2 00 - 11757), du 26 février 2016? NON V3Q3

# Cast-as-Intended Verification

- Detectable malware attacks
  - Manipulated votes ✓
  - Suppressed votes ✓
  - Manipulated verification codes ✓
  - Suppressed verification codes ✓
- Unsolved malware attacks
  - Secrecy of vote ×
  - Social engineering attack: “Please enter verification code” ×

Bern University
of Applied Sciences

# CHVote Voting Protocol

# CHVote Project

- Project goals
  - New implementation from scratch
  - Reach second expansion stage in one step (100% electorate)
  - Developed, hosted, operated entirely by the State of Geneva
- Strategy
  - Collaboration with academia (BFH, EPFL, LORIA, Bristol)
  - State-of-the-art technologies
  - Maximal transparency
  - High-quality open documentation
  - Open-source license (AGPL 3.0)
  - Invitation to public code reviewing
- Scheduled to be used for Federal Council elections in 2019

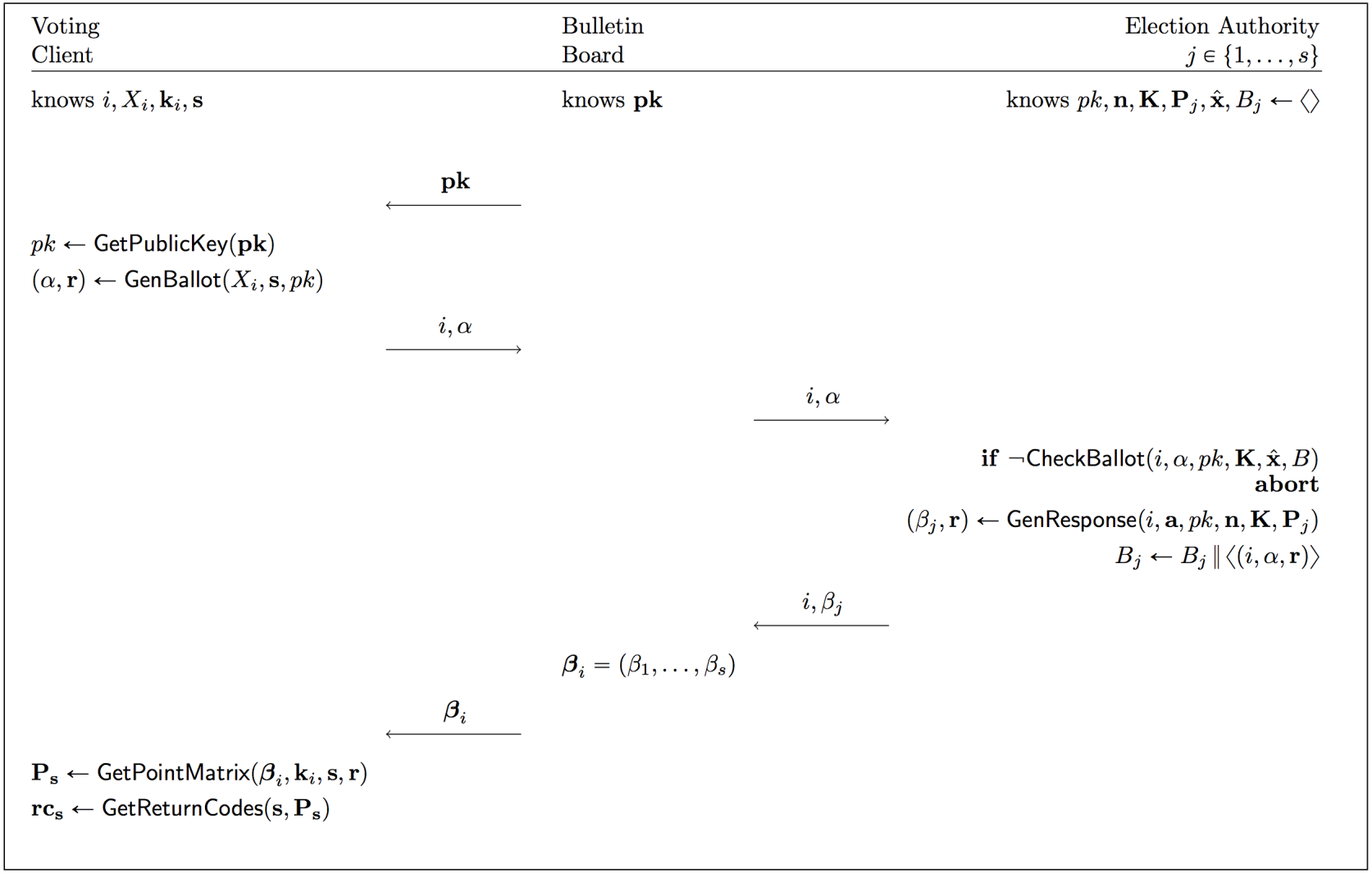
# CHVote Voting Protocol

- Key cryptographic ingredients
  - Schnorr identification scheme
  - Distributed generation of credentials and verification codes
  - Oblivious transfer of selected codes
  - Verifiable re-encryption mix-net
  - Distributed decryption with shared ElGamal private key
- Trust assumptions
  - Honest printing authority and postal mail
  - At least one honest election authority (for vote integrity)
  - Polynomially-bounded adversary
  - Decisional Diffie-Hellaman problem is hard
  - No “family voting”, no vote buying, no coercion
  - No privacy attacks on voting client

# CHVote Protocol Specification

- Published on April 20, 2017
- Self-contained and comprehensive document (∼140 pages)
  - Description of election use cases
  - Mathematical and cryptographic background
  - Details of encoding and hashing algorithms
  - Adversary and trust assumptions
  - Cryptographic and election parameters
  - Recommendations for group sizes, key lengths, code lengths
- Three main protocols: pre-election ⇒ election ⇒ post-election
- About 60 pseudo-code algorithms
- Scientific papers presented at E-Vote-ID'16, FC'17, FC'18

| Phase | Election Admin. | Election Authority | Printing Authority | Voter | Voting Client | Bulletin Board | Protocol Nr. |
|---|---|---|---|---|---|---|---|
| 1. Pre-Election | • | • | • | • | | • | |
| 1.1 Election Preparation | • | • | | | | • | 6.1 |
| 1.2 Printing of Code Sheets | | • | • | • | | • | 6.2 |
| 1.3 Key Generation | | • | | | | • | 6.3 |
| 2. Election | | • | | • | • | • | |
| 2.1 Candidate Selection | | | | • | • | • | 6.4 |
| 2.2 Vote Casting | | • | | | • | • | 6.5 |
| 2.3 Vote Confirmation | | • | | • | • | • | 6.6 |
| 3. Post-Election | • | • | | | | • | |
| 3.1 Mixing | | • | | | | • | 6.7 |
| 3.2 Decryption | | • | | | | • | 6.8 |
| 3.3 Tallying | • | | | | | • | 6.9 |




Protocol 6.5: Vote Casting

**Algorithm:** GenBallot$(X, \mathbf{s}, pk)$

**Input:** Voting code $X \in A_X^{\ell_X}$
Selection $\mathbf{s} = (s_1, \ldots, s_k)$, $1 \leqslant s_1 < \cdots < s_k$
Encryption key $pk \in \mathbb{G}_q \backslash \{1\}$

$x \leftarrow$ ToInteger$(X)$ // see Alg. 4.7
$\hat{x} \leftarrow \hat{g}^x \bmod \hat{p}$
$\mathbf{q} \leftarrow$ GetSelectedPrimes$(\mathbf{s})$ // $\mathbf{q} = (q_1, \ldots, q_k)$, see Alg. 7.19
$m \leftarrow \prod_{i=1}^{k} q_i$
**if** $m \geqslant p$ **then**
  **return** $\perp$ // $(k, n)$ is incompatible with $p$
$(\mathbf{a}, \mathbf{r}) \leftarrow$ GenQuery$(\mathbf{q}, pk)$ // $\mathbf{a} = (a_1, \ldots, a_k)$, $\mathbf{r} = (r_1, \ldots, r_k)$, see Alg. 7.20
$a \leftarrow \prod_{i=1}^{k} a_i \bmod p$
$r \leftarrow \sum_{i=1}^{k} r_i \bmod q$
$b \leftarrow g^r \bmod p$
$\pi \leftarrow$ GenBallotProof$(x, m, r, \hat{x}, a, b, pk)$ // $\pi = (t, s)$, see Alg. 7.21
$\alpha \leftarrow (\hat{x}, \mathbf{a}, b, \pi)$
**return** $(\alpha, \mathbf{r})$ // $\alpha \in \mathbb{Z}_{\hat{q}} \times \mathbb{G}_q^k \times \mathbb{G}_q \times ((\mathbb{G}_{\hat{q}} \times \mathbb{G}_q^2) \times (\mathbb{Z}_{\hat{q}} \times \mathbb{G}_q \times \mathbb{Z}_q))$, $\mathbf{r} \in \mathbb{Z}_q^k$

Algorithm 7.18: Generates a ballot based on the selection $\mathbf{s}$ and the voting code $X$.

```java
/**
 * Algorithm 7.18: GenBallot
 *
 * @param upper_x the voting code
 * @param bold_s  voters selection (indices)
 * @param pk      the public encryption key
 * @return the combined ballot, OT query and random elements used
 */
public BallotQueryAndRand genBallot(String upper_x, List<Integer> bold_s, EncryptionPublicKey pk) {
    BigInteger x = conversion.toInteger(upper_x, publicParameters.getUpper_a_x());
    BigInteger x_circ = modExp(g_circ, x, p_circ);
    List<BigInteger> bold_q = computeBoldQ(bold_s);
    BigInteger m = computeM(bold_q, p);
    ObliviousTransferQuery query = genQuery(bold_q, pk);
    BigInteger a = computeA(query, p);
    BigInteger r = computeR(query, q);
    BigInteger b = modExp(g, r, p);
    NonInteractiveZKP pi = genBallotProof(x, m, r, x_circ, a, b, pk);
    BallotAndQuery alpha = new BallotAndQuery(x_circ, query.getBold_a(), b, pi);

    return new BallotQueryAndRand(alpha, query.getBold_r());
}
```

https://github.com/republique-et-canton-de-geneve/chvote-protocol-poc

# Crypto-Algorithms in Pseudo-Code

- Ideal interface between cryptographers, developers, auditors
  - Cryptographers can write, read, and check pseudo-code
  - Developers can derive real code from pseudo-code
  - Auditors can check if pseudo-code and real code match
  - Useful for security proofs
- Rarely used in ...
  - cryptographic literature
  - electronic voting protocols
- Often used in standards (FIPS, RFC, PKCS, ...)

# Conclusion

# Conclusion

- Verifiability is central to making e-voting secure
- The CHVote is project is on the right track (transparency, free software license, open documentation, academic partners)
- The specification document is one of the most detailed and comprehensive in the world
- Proof-of-concept code exists in Java and Phyton
- Suitable for GI elections?

# Challenges and Open Problems

- Complexity of cryptographic protocols
- Cryptography in web browser (JavaScript)
- Vote secrecy on insecure platform
- Vote buying and coercion
- Everlasting privacy

# Links and Demo

- Specification document
  - https://eprint.iacr.org/2017/325
- Proof-of-concept implementation (Java)
  - https://github.com/republique-et-canton-de-geneve/chvote-protocol-poc
- Bachelor thesis by Y. Denzer and K. Häni (January 2018)
  - One-to-one implementation of CHVote specification
  - Made for educational purpose only
  - Demo available at https://chvote.ti.bfh.ch
  - Python code available at https://github.com/nextgenevoting