Security and Privacy

sVote protocol

28.05.2019

Ph. Oechslin

EPFL

Svote protocol

- Agenda
  - History and main facts
  - Design, architecture
  - Cryptographic protocol
  - Mixing and decryption

Main facts

Svote protocol

- History
  - Developed by Spanish company Scytl
  - First used by Neuchâtel
  - Now run by Swiss Post for various cantons
- Main facts
  - End-to-end encryption
  - Encrypted ballot is cast into ballot box
  - Blinded ballot is cast to verification code generator
  - NIZKPs are used to prove that the encryption and the blinded ballot contain the same vote
  - Mixes reencrypt for anonymity and perform partial decryption while mixing

Svote protocol

- Main codes
  - Initialization code is used to login (possibly with a birth year)
  - Verification codes to verify that votes were transmitted as cast
  - Confirmation code to confirm the vote
  - Finalisation code to confirm that the vote is cast called Vote Cast Code

Cryptographic Protocol

Svote

Authentication

- The voter enters their initialization code to log in.
- Two different keys are derived from the code
  - the voting card id is used in an URL to download an encrypted file (keystore)
  - the startvoting key is used to decrypt that keystore
- The key store contains all keys and parameters needed by the voter

Voter: Encrypted vote

- vote options (candidates) v_i are represented with small prime numbers $\in \mathbb{G}_q$
- before encryption, the t selected options are multiplied together and encrypted with a single ElGamal encryption
  - this is efficient for encryption, mixing and decryption
  - after decryption, the options can be recovered through factorization

  $v = \prod_{l=1}^{t}(v_l)$

  $(c_1, c_2) = \texttt{Enc}_{pk}(v) = (v \cdot pk^r, g^r)$
- Votes from all voters are encrypted with the same public key, but with different randomness r

Voter: Verification codes

- verification codes are generated from the options through exponentiation with secret keys that are unique to each voter (blinding).
- The client applies their secret exponent, the voting card private key $\mathtt{VC}_{sk}^{id}$ to each vote option, resulting in the partial verification codes $\{\mathtt{pvc}_l^{id}\}_{l=1}^{t} = (v_1^{\mathtt{VC}_{sk}^{id}}, \ldots, v_t^{\mathtt{VC}_{sk}^{id}})$
- The public key of this exponentiation is $\mathtt{VC}_{pk}^{id} = g^{\mathtt{VC}_{sk}^{id}}$
- The partial verification codes will be used by the server to obtain the verification codes with help of the control components

Proofs

- How do we know that the encrypted vote and the partial verification codes correspond to the same vote ?
- We need one intermediate value (cipher text exponentiations) and a few proofs to demonstrate this.
- Cipher text exponentiations
  - We take the encrypted vote and exponentiate it with the same key as the partial verification codes: $(\tilde{c}_1, \tilde{c}_2) = (c_1^{\mathtt{VC}_{sk}^{id}}, c_2^{\mathtt{VC}_{sk}^{id}})$

Proofs

- Schnorr Proof: Prove knowledge of r in encryption of ballot, bind the proof to the voting card id:

$$\pi_{schnorr} = NIZKP[(r) : c_2 = g^r, 'voterID = id']$$

- Proof of exponentiation: Proof that we correctly calculated the exponentiations of c_1 and c_2.
  - The VC public key and $\tilde{c}_1$,$\tilde{c}_2$ have the same logarithm (the VC private key):

$$\pi_{exp} = NIZKP[(\mathtt{VC}^{id}_{sk}) : \mathtt{VC}^{id}_{pk} = g^{\mathtt{VC}^{id}_{sk}} \wedge \tilde{c}_1 = c_1^{\mathtt{VC}^{id}_{sk}} \wedge \tilde{c}_2 = c_2^{\mathtt{VC}^{id}_{sk}}]$$

Proofs

- Plaintext equivalence proof: Proof that the encrypted vote and the partial choice codes contain the same options.
  - If it is true, they cancel out if we divide $\tilde{c}_2$ by the product of the partial choice codes:

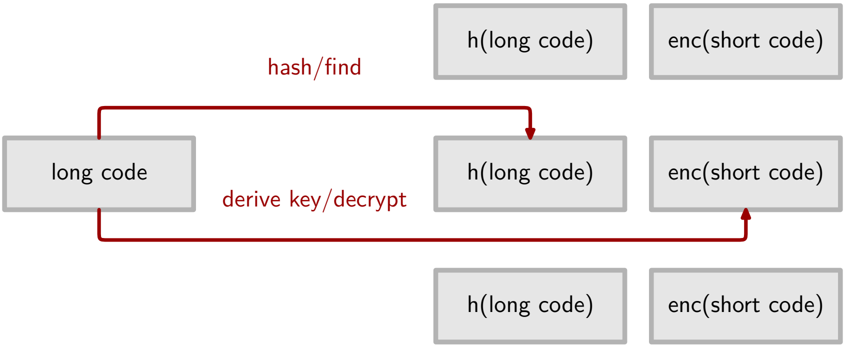
$$\frac{\tilde{c}_1}{\prod_{l=1}^{t} \mathtt{pvc}_l^{id}} = \frac{(c_1)^{\mathtt{VC}_{sk}^{id}}}{\prod_{l=1}^{t} v_l^{\mathtt{VC}_{sk}^{id}}} = \frac{\prod_{l=1}^{t} v_l^{\mathtt{VC}_{sk}^{id}} (pk^r)^{\mathtt{VC}_{sk}^{id}}}{\prod_{l=1}^{t} v_l^{\mathtt{VC}_{sk}^{id}}} = (pk^r)^{\mathtt{VC}_{sk}^{id}}$$

  - we have already proven that $\tilde{c}_2$ has this exponent: $\tilde{c}_2 = (g^r)^{\mathtt{VC}_{sk}^{id}}$
  - proof that both exponents are the same:

$$\pi_{pleq} = NIZKP[(r \cdot \mathtt{VC}_{sk}^{id}) : \tilde{c}_2 = g^{r\mathtt{VC}_{sk}^{id}} \wedge \frac{\tilde{c}_1}{\prod_{l=1}^{t} \mathtt{pvc}_l^{id}} = pk^{r\mathtt{VC}_{sk}^{id}}]$$

Generation of short code

- The partial verification codes (v^k) are hashed by the server with the help of the CCs to obtain a long verification code
- With key and a table the server maps long codes to short ones



Mixing

Mixing

- **Architecture**
  - 3 CCs at Swiss Post plus one CC at the canton are used
  - Swiss Post is thus not able to break vote secrecy during decryption and mixing
- **Operations**
  - Cleansing: invalid (e.g. unconfirmed) ballots are removed by the server
  - Mix/decrypt: the ballots are
    - re-encrypted
    - shuffled
    - The method used is called Bayer-Groth
      - It generates proofs that no vote has been modified
  - Partial decryption: each mixer decrypts with its key share
    - An NIZKP is generated to proof that we decrypted with the correct private key
    - after the last mix, the votes are in cleartext

Conclusions

- for each vote we receive
  - Schnorr proof: we have a proof that it was generated with a valid voting card
    - we have proof that each voting card was used only once
  - Exponentiation proof + Plaintext equality proof: we have proof that the correct verification codes were generated
  - We obtained the confirmation code from the voter: proof that they agree with the verification code
  - Shuffle proofs: we have proof that during mixing the content of the votes was not modified
  - Decryption proof: we have proof the clear text is the correct decryption of the cipher text

References

- The details of Svote and it source code have been published by Swiss Post in **Gitlab**
  - you need to accept their terms and conditions before you get access
- The **cryptographic** protocol, and a proof that it is correct is published on the **web site of swiss Post**