Bern University
of Applied Sciences

Cast-as-Intended Verification in Electronic Elections Based on Oblivious Transfer

Rolf Haenni, Reto Koenig, Eric Dubuis

Bregenz, October 21st, 2016

Table of Contents

Introduction

Cast-As-Intended Verification

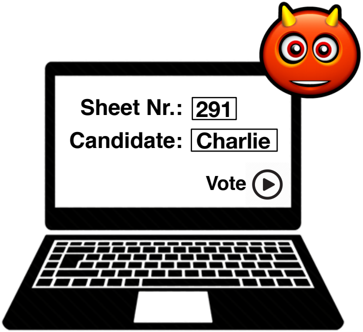
Code Sheet Nr. 291

Candidates	Codes
Alice	7449
Bob	8243
Charlie	9123

Cast-As-Intended Verification

Code Sheet Nr. 291

Candidates	**Codes**
Alice	7449
Bob	8243
Charlie	9123



Cast-As-Intended Verification

Code Sheet Nr. 291

Candidates	**Codes**
Alice	7449
Bob	8243
Charlie	9123

Cast-As-Intended Verification

Code Sheet Nr. 291

Candidates	Codes
Alice	7449
Bob	8243
Charlie	9123

Existing Work I

Gjøsteen (VoteID'11), Puiggalí, Guasch (VoteID'11, EVOTE'12)

- Based on blinding the encrypted vote
- Implemented and tested in Norway
- Return codes transmitted by SMS
- Vote updating allowed

Galindo, Guasch, Puiggalí (VoteID'15)

- Implemented and tested in Neuchâtel (Switzerland)
- Return codes displayed on voting platform
- Vote updating not allowed

Existing Work II

Heiberg, Lipmaa, van Laenen (ESORICS'10)

- Based on proxy oblivious transfer
- Return codes transmitted over "postchannel"
- Vote updating allowed
- Well-formedness of ballot based on range proofs
- Only for 1-out-of-n elections
- Generalization to k-out-of-n elections very inefficient
- Inspiration for this paper

Motivation

- Security properties of transmitting return codes
  1. The voting server does not learn the voter's selections
  2. The voting platform does not learn codes different from the voter's selections
- In cryptography, this is called an oblivious transfer (OT) problem
- Motivation: Study existing OT schemes and apply them to cast-as-intended verification

Oblivious Transfer

Oblivious Transfer: k-out-of-n

- A k-out-of-n oblivious transfer is a protocol between a sender and a receiver
  - The sender has n messages $\mathbf{m} = (m_1, \ldots, m_n)$, $m_i \in \{0,1\}^\ell$
  - The receiver selects k indices $\mathbf{s} = (s_1, \ldots, s_k)$, $s_i \in \{1, \ldots, n\}$
  - Executing the protocol reveals $\mathbf{m}_\mathbf{s} = (m_{s_1}, \ldots, m_{s_k})$ to the receiver

- Receiver privacy: the sender learns nothing about $\mathbf{s}$
- Sender privacy: the receiver learns nothing about $\mathbf{m}$ other than $\mathbf{m}_\mathbf{s}$

OT-Scheme by Chu and Tzeng

- C. Chu and W. Tzeng, *Efficient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries*, Workshop on Theory and Practice in Public Key Cryptography, 2005
- Their OT scheme consists of three algorithms:

$$\mathbf{a} \leftarrow \text{Query}(\mathbf{s}, \mathbf{r})$$
$$(\mathbf{b}, \mathbf{c}, d) \leftarrow \text{Response}(\mathbf{a}, \mathbf{m}, r)$$
$$\mathbf{m_s} \leftarrow \text{Open}(\mathbf{b}, \mathbf{c}, d, \mathbf{r})$$

satisfying $\mathbf{m_s} = \text{Open}(\text{Response}(\text{Query}(\mathbf{s}, \mathbf{r}), \mathbf{m}, r), \mathbf{r})$ for all possible inputs $\mathbf{s}$, $\mathbf{m}$, $\mathbf{r}$, and r

Public Parameters

- Subgroup $\mathbb{G} \subset \mathbb{Z}_p^*$ of integers modulo $p = 2q + 1$
- Generator $g \in \mathbb{G} \setminus \{1\}$
- $\Gamma : \{1, \ldots, n\} \to \mathbb{G}$
- Message length ℓ
- Colllision-resistant hash function $H : \mathbb{G} \to \{0, 1\}^\ell$

Protocol Description

Receiver
selects $\mathbf{s} = (s_1, \ldots, s_k)$

for $j = 1, \ldots, k$
– pick random $r_j \in_R \mathbb{Z}_q$
– compute $a_j = \Gamma(s_j) \cdot g^{r_j}$

$\xrightarrow{\mathbf{a}=(a_1,\ldots,a_k)}$

Sender
knows $\mathbf{m} = (m_1, \ldots, m_n)$

pick random $r \in_R \mathbb{Z}_q$
for $j = 1, \ldots, k$
– compute $b_j = a_j^r$
for $i = 1, \ldots, n$
– compute $k_i = H(\Gamma(i)^r)$
– compute $c_i = m_i \oplus k_i$
compute $d = g^r$

$\mathbf{b}=(b_1,\ldots,b_k)$
$\mathbf{c}=(c_1,\ldots,c_n)$
d
$\longleftarrow$

for $j = 1, \ldots, k$
– compute $k_j = H(b_j \cdot d^{-r_j})$
– compute $m_{s_j} = c_{s_j} \oplus k_j$

Properties

- Provably secure against malicious receiver and semi-honest sender (in the random oracle model)
  - Unconditional receiver privacy
  - Computational sender privacy under the CDH assumption
- Performance
  - Receiver: $2k$ exponentiations (k can be precomputed)
  - Sender: $n+k+1$ exponentiations ($n+1$ can be precomputed)
  - Hence k online exponentiations for both sender and receiver
- Compatible with ElGamal encryption
  - Replace g by public key $pk = g^{sk}$
  - $a_j = \Gamma(s_j) \cdot pk^{r_j}$ is left-hand side of ElGamal ciphertext:

$$(a_j, g^{r_j}) = Enc_{pk}(\Gamma(s_j), r_j)$$

Overview of Vote Casting Process

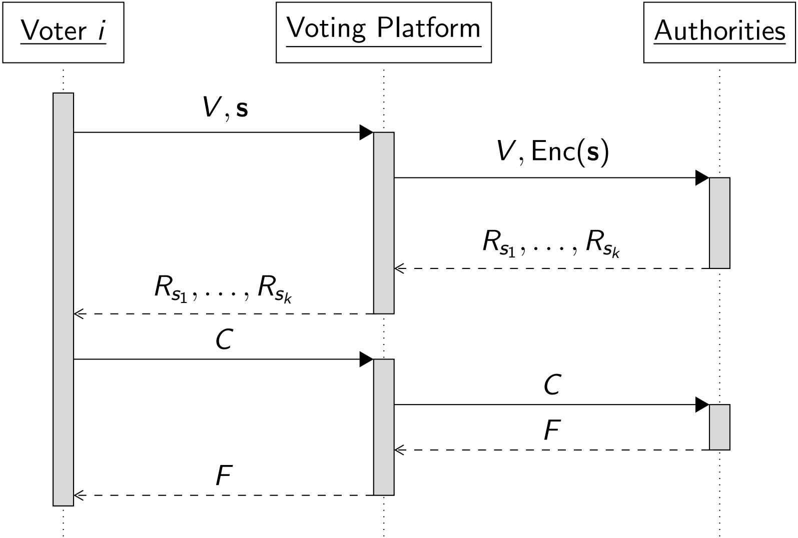
General Setting

- Involved parties
  - Voters
  - Untrusted voting platforms
  - Trusted authorities (one or multiple)
- Communication over broadcast channel with memory (public bulletin board)
- Number of candidates n
- Number of selections $k \leq n$
- Voter's selection $\mathbf{s} = (s_1, \ldots, s_k)$, for $1 \leq s_1 < \cdots < s_k \leq n$

Code Sheets

- Code sheets are generated by the authorities and sent to voters by trusted postal mail
- Personalized information printed code sheets:
  - Voting code V
  - Return codes $R_1, \ldots, R_n$ (next to candidate names)
  - Confirmation code C
  - Finalization code F
- We assume that voter authentication is solved, i.e., possession of code sheet implies eligibility

Simplified Voting Protocol



Main Challenges

- Transfer return codes $R_{s_1}, \ldots, R_{s_k}$ to voter such that ...
  - the authorities learn nothing about $\mathbf{s}$
  - the voting platform learns no other R_j for $j \notin \mathbf{s}$
- Convince authorities that Enc($\mathbf{s}$) contains a valid vote:
  - $\mathbf{s} = (s_1, \ldots, s_k)$
  - $s_i \in \{1, \ldots, n\}$
  - $s_i \neq s_j$

  Usually implemented with non-interactive zero-knowledge proofs (by Groth, Lipmaa, Joaquim, etc.)

Protocol Description

Public parameters

- Subgroup $\mathbb{G} \subset \mathbb{Z}_p^*$ of integers modulo $p = 2q + 1$
- Public ElGamal key $pk \in \mathbb{G} \setminus \{1\}$
- $\Gamma(i) = p_i$, where p_i is the i-th prime number in $\mathbb{G}$
- $\Gamma(\mathbf{s}) = \prod_{j=1}^{k} \Gamma(s_j) = \prod_{j=1}^{k} p_{s_j}$ for $\mathbf{s} = (s_1, \ldots, s_k)$
- Note that decoding $\mathbf{s} = \Gamma^{-1}(\Gamma(\mathbf{s}))$ is efficient by factorization

Vote Casting

- The voter selects $\mathbf{s} = (s_1, \ldots, s_k)$ and enters $V, \mathbf{s}$ into the voting platform
- The voting platform performs the following steps:
  - Pick random values $\mathbf{r} = (r_1, \ldots, r_k) \in_R \mathbb{Z}_q^k$
  - Compute $\mathbf{a} = (a_1, \ldots, a_k) = \text{Query}(\mathbf{s}, \mathbf{r})$
  - Compute $b = g^r$ for $r = \sum_{j=1}^k r_j$
  - Compute NIZKP π that proves knowledge of r
  - Send ballot $\beta = (V, \mathbf{a}, b, \pi)$ to authorities

- Note that $(a, b) = \text{Enc}_{pk}(\Gamma(\mathbf{s}), r)$, where

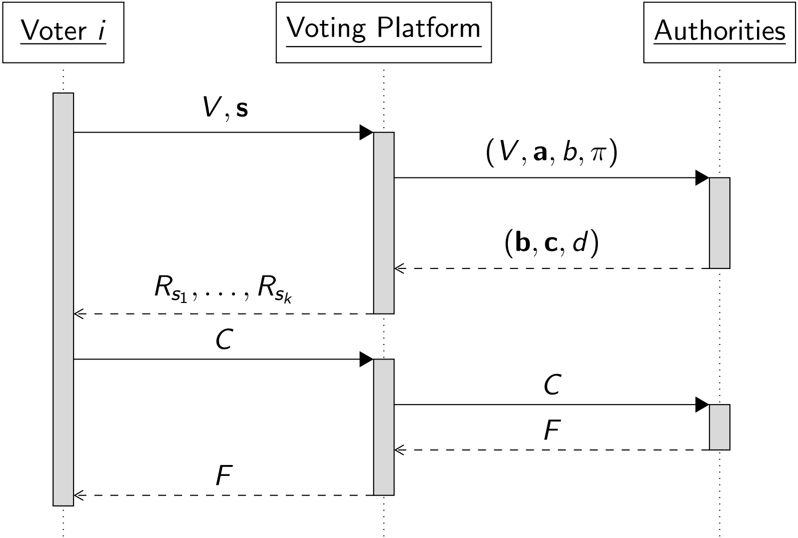
  $$a = \prod_{j=1}^k a_j = \prod_{j=1}^k \Gamma(s_j) \cdot pk^{r_j} = \Gamma(\mathbf{s}) \cdot pk^{\sum_{j=1}^k r_j} = \Gamma(\mathbf{s}) \cdot pk^r,$$

  is an ElGamal ciphertext

Vote Confirmation

- Let $\mathbf{m} = (R_1, \ldots, R_n)$ be the return codes for a given voter
- The authorities perform the following steps:
  - Check V and π, abort in case checks fail
  - Pick random value $r \in_R \mathbb{Z}_q$
  - Compute $(\mathbf{b}, \mathbf{c}, d) = \mathsf{Response}(\mathbf{a}, \mathbf{m}, r)$
  - Send response $(\mathbf{b}, \mathbf{c}, d)$ to voting platform
- The voting platform gets $(R_{s_1}, \ldots, R_{s_k}) = \mathsf{Open}(\mathbf{b}, \mathbf{c}, d, \mathbf{r})$ and displays them to the voter
- If the displayed return codes match with the code sheet, the voter enter the confirmation code C to the voting platform, which sends C to the authorities
- If the confirmation code is correct, the authorities respond with the finalization code F

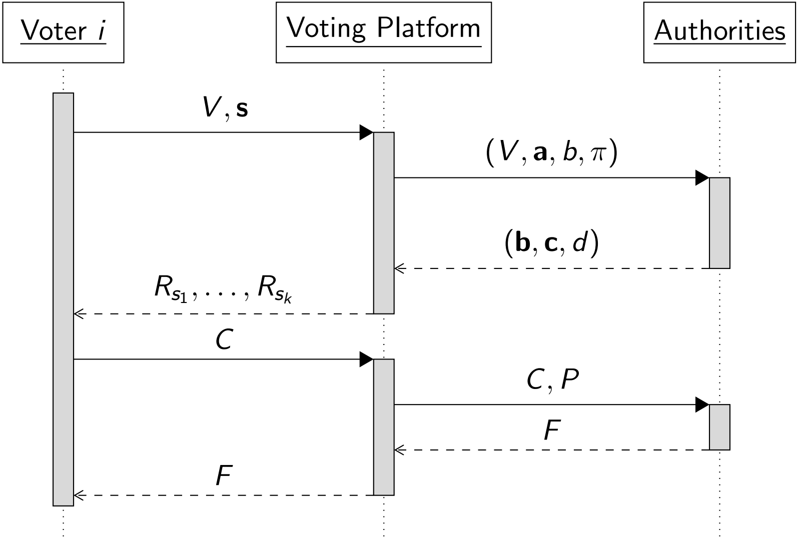
Less Simplified Voting Protocol



Vote Validation

- Convince authorities that $(a, b) = \text{Enc}_{pk}(\Gamma(\mathbf{s}), r)$ contains valid vote $\mathbf{s}$
- The authorities generate return codes $(R_1, \ldots, R_n)$ as follows:
  - Pick random polynomial $p(X) = \sum_{i=0}^{k-1} a_i X^i$ of degree $k-1$
  - Compute n points $p_i = (x_i, p(x_i))$ for random $x \in_R \mathbb{Z}_p$
  - Let $R_i = \text{Hash}(p_i)$
- Transfer $\mathbf{m_s} = (p_{s_1}, \ldots, p_{s_k})$ obliviously to voting platform
- The voting platform performs the following steps:
  - Compute $R_{s_i} = \text{Hash}(p_{s_i})$ and display them to the voter
  - Interpolate $p(X)$ from $\mathbf{m_s}$ and compute $P = p(0)$
  - Send (C, P) to confirm vote
- Only if the both C and P are correct, the authorities respond with the finalization code F

Voting Protocol



Discussion and Conclusion

Performance

		This Paper	Scytl (VoteID'15)
Preparation	Authorities	$6N$	$(n+2)N$
Vote Casting	Voting platform	$2k+3$ $(k+3)$	$k+10$ (7)
	Authorities	$n+k+5$ $(n+1)$	11 (0)

Conclusion

- Compared to existing approaches, our protocol . . .
  - provides a conceptually more elegant solution based on OT
  - includes no cryptographic keys other than pk
  - requires a single type of authority
  - is based on weaker trust assumptions
  - is equally efficient
- The protocol generalizes naturally to multiple authorities and to multiple (parallel) k-out-of-n elections
- The protocol is compatible with the requirements of the Swiss Federal Chancellery
- The Canton of Geneva plans to implement it in CHvote

Questions?

Source: https://en.wikipedia.org/wiki/Landsgemeinde