Towards a Mechanized Proof of Selene Receipt-Freeness and Vote-Privacy

Alessandro Bruni, Eva Drewsen, Carsten Schürmann

IT UNIVERSITY OF COPENHAGEN

E-Vote-ID 2017

Selling Formal Verification

"Security protocols are three line programs that people still manage to get wrong"

(Roger Needham)

Selling Formal Verification

"Security protocols are three line programs that people still manage to get wrong"

(Roger Needham)

Voting protocols are ten line programs that:

- use hard crypto

  homomorphic encryption, zero-knowledge proofs, commitment schemes, oblivious transfers, threshold cryptography...
- have colorful security properties

  vote-privacy, individual verifiability, universal verifiability, coercion resistance, receipt freeness...
- *sometimes* do not come with security proofs

Selling Formal Verification

"Security protocols are three line programs that people still manage to get wrong"

(Roger Needham)

Voting protocols are ten line programs that:

- use hard crypto
  homomorphic encryption, zero-knowledge proofs, commitment schemes, oblivious transfers, threshold cryptography...
- have colorful security properties
  vote-privacy, individual verifiability, universal verifiability, coercion resistance, receipt freeness...
- *sometimes* do not come with security proofs
- (and *kill your favourite verification engine*)

Contributions

We propose:

- The first formal model of the Selene voting protocol
- A simplified version of the protocol amenable to automatic verification
- A convergent equational theory for Pedersen-style commitments used by Selene

Results:

- We prove Vote Privacy in our model
- We show a known attack for Selene Receipt Freeness, and prove the security of the corrected version

The Selene E-voting Protocol

- Due to Peter Ryan, Peter Rønne, Vincenzo Iovino
- Internet voting protocol designed for low-coercion scenarios

The Selene E-voting Protocol

- Due to Peter Ryan, Peter Rønne, Vincenzo Iovino
- Internet voting protocol designed for low-coercion scenarios

Key ideas

1. votes are publicly posted on a bulletin board makes it easy to trust the result;
2. tracking receipts (*tracker numbers*) allow users to trust that their vote has been cast, ✓individual verifiability
3. and to fake receipts for potential coercers. ✓receipt freeness

El-gamal cryptosystem

Gen: Select a subgroup $G \subset \mathbb{Z}_p^*$ of order q, and a generator g of G. Choose $x \underset{R}{\leftarrow} Z_q$. Reveal $h = g^x$.

Enc: To encrypt a message $m \in G$, we choose $r \underset{R}{\leftarrow} Z_q$. The ciphertext is then:

$$(c, d) = (g^r, m \cdot h^r).$$

Dec: To decrypt the ciphertext (c, d), compute

$$m = \frac{d}{c^x}.$$

El-gamal homomorphisms:

Reencryption:

Let $(g^r, m \cdot h^r)$ be an encryption of m with randomness r.

By chosing $r' \underset{R}{\leftarrow} Z_q$, we can re-encrypt the message m with $(g^{r+r'}, m \cdot h^{r+r'}) = (g^r, m \cdot h^r) \cdot (g^{r'}, 1 \cdot h^{r'})$.

Shuffling mixnets can be built by chaining re-encryption mixers, that apply re-encryption and randomly shuffle the values.

If at least one node in the mixnet is honest, the link between input and output is lost from the perspective of an observer.

Pedersen-style commitment

Gen: Select a subgroup $G \subset \mathbb{Z}_p^*$ of order q, and a generator g of G. Choose $x \underset{R}{\leftarrow} Z_q$. Reveal $h = g^x$.

Pedersen-style commitment

Gen: Select a subgroup $G \subset \mathbb{Z}_p^*$ of order q, and a generator g of G. Choose $x \underset{R}{\leftarrow} Z_q$. Reveal $h = g^x$.

Commit: To commit to a message $m \in G$, we choose $r \underset{R}{\leftarrow} Z_q$. The commitment is then: $c = g^m \cdot h^r$.

Pedersen-style commitment

Gen: Select a subgroup $G \subset \mathbb{Z}_p^*$ of order q, and a generator g of G. Choose $x \underset{R}{\leftarrow} Z_q$. Reveal $h = g^x$.

Commit: To commit to a message $m \in G$, we choose $r \underset{R}{\leftarrow} Z_q$. The commitment is then: $c = g^m \cdot h^r$.

Open: To reveal the message m, the second component is sent: $d = g^r$.

Pedersen-style commitment

Gen: Select a subgroup $G \subset \mathbb{Z}_p^*$ of order q, and a generator g of G. Choose $x \underset{R}{\leftarrow} Z_q$. Reveal $h = g^x$.

Commit: To commit to a message $m \in G$, we choose $r \underset{R}{\leftarrow} Z_q$. The commitment is then: $c = g^m \cdot h^r$.

Open: To reveal the message m, the second component is sent: $d = g^r$.

Properties

Information theoretically hiding: given the commitment c, any message $m' \in G$ is equally likely, and in particular, having the secret key x one can compute: $r' = \frac{m-m'}{x} + r$

Pedersen-style commitment

Gen: Select a subgroup $G \subset \mathbb{Z}_p^*$ of order q, and a generator g of G. Choose $x \underset{R}{\leftarrow} Z_q$. Reveal $h = g^x$.

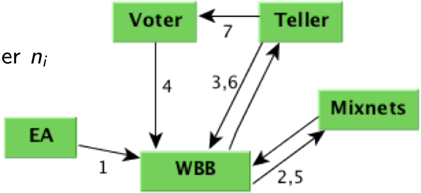
Commit: To commit to a message $m \in G$, we choose $r \underset{R}{\leftarrow} Z_q$. The commitment is then: $c = g^m \cdot h^r$.

Open: To reveal the message m, the second component is sent: $d = g^r$.

Properties

Information theoretically hiding: given the commitment c, any message $m' \in G$ is equally likely, and in particular, having the secret key x one can compute: $r' = \frac{m-m'}{x} + r$
Computationally binding: finding two messages m and m' that open the commitment c requires finding an r and r' s.t. $g^m \cdot h^r = g^{m'} \cdot h^{r'}$; then one can compute $\log_g(h) = \frac{m'-m}{r-r'}$.

Selene: Voting in seven "easy" steps



1. Election Authority produces a tracker number n_i and its encryption e_i for each Voter i;
2. Mixnet shuffles the encrypted trackers e_i, resulting in a re-encryption e_i' that loses connection to n_i;
3. Teller(s) generate Pedersen commitments c_is for the n_is, assign them to Voters V_i, then publish them to the Bulletin Board
4. Votes v_i are encrypted (ev_i) and signed (s_i) by Voters V_i, and published along
5. Encrypted tracking numbers and votes $\langle e_i', ev_i \rangle$ are shuffled by the Mixnet, then published as $\langle e_i'', ev_i' \rangle$, losing link to the originals;
6. Votes ev_i and trackers c_is are decrypted by the Tellers, and published to the Bulletin Board
7. Commitments are opened by the Teller(s) to the Voters, who can check that their vote has been casted.

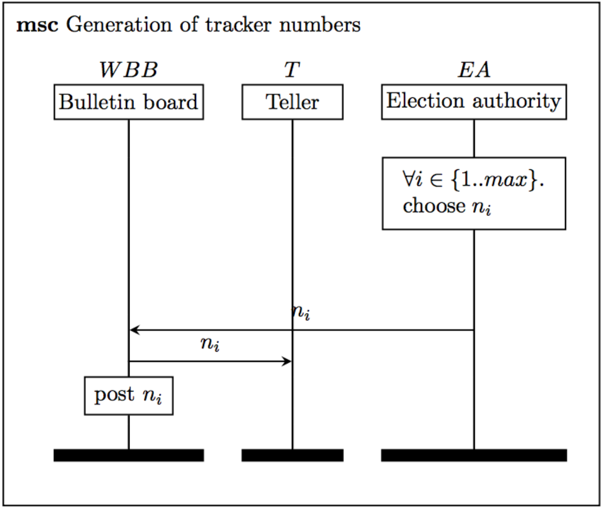
Caveat emptor

To allow mechanised analysis in Tamarin, we assume an *external active adversary* who may collude with one of two voters.

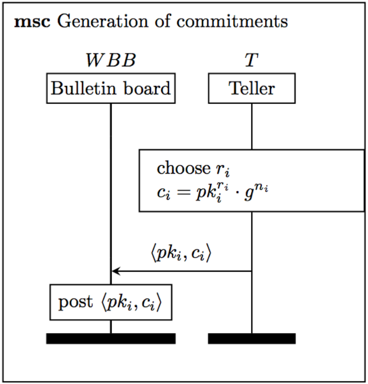
Therefore:

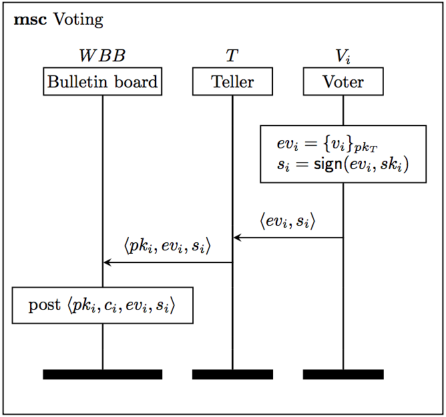
- Only one teller is needed (since we assume it is honest)
- Re-encryption mixing is replaced by ballot shuffling
- No zero-knowledge proofs of secure computation are needed for the Teller, Bulletin Board and Election Authority
- We assume the existence of authentic and confidential channels for all communication between the honest parties

Simplified Selene (1)

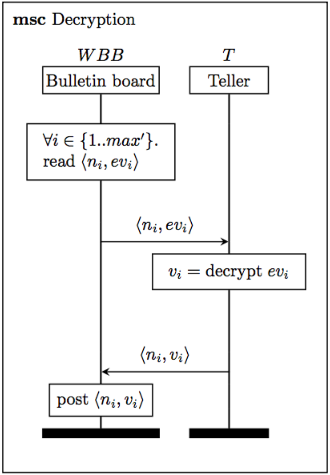


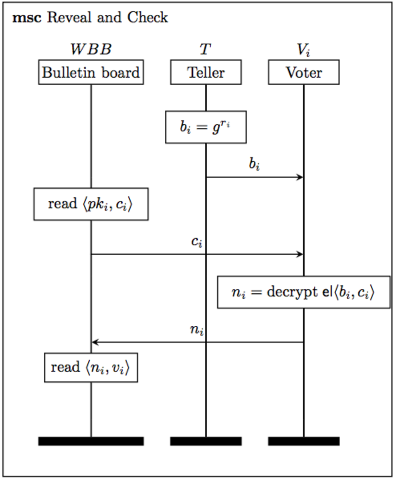
Simplified Selene (2)





Simplified Selene (3)





Tamarin 101

Theorem prover based on multiset-rewrite rules:

$$l \text{-}[\alpha]\!\rightarrow r$$

- States S are multiset of facts (initial state $\emptyset$)
- Bang (!) modality for replicated facts
- Given a rule $l \text{-}[\alpha]\!\rightarrow r$ and a substitution σ, a transition $\sigma(l \xrightarrow{\alpha} r)$ can fire on state S iff $\sigma(l) \subseteq S$, and produces a state $S' = S \setminus \alpha(l \uplus r)$

Tamarin 101

Theorem prover based on multiset-rewrite rules:

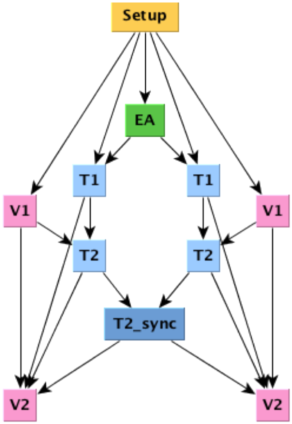
$$l \text{-}[\alpha]\!\rightarrow r$$

- States S are multiset of facts (initial state $\emptyset$)
- Bang (!) modality for replicated facts
- Given a rule $l \text{-}[\alpha]\!\rightarrow r$ and a substitution σ, a transition $\sigma(l \xrightarrow{\alpha} r)$ can fire on state S iff $\sigma(l) \subseteq S$, and produces a state $S' = S \setminus \alpha(l \uplus r)$

Observational equivalence "Given two systems, equivalent rules should fire in equivalent states."

$\mathit{diff}(t_1, t_2)$ terms are used to distinguish the two systems

Tamarin model



Rules:

- *Setup*, generates key pairs, initializes all agents
- *EA*, Election Authority, generates tracker numbers
- $T1$, teller commits a tracker number to each voter
- $V1$, voting phase
- $T2$, teller decrypts the encrypted vote
- $T2_{sync}$, all votes are shuffled
- $V2$, "reveal and check"

Equational theory

To model the commitment scheme we need the following two equations:

1. $open(commit(n, r, pk(sk)), r, sk) = n$
2. $commit(n_2, fake(n_1, r, sk, n_2), pk(sk)) = commit(n_1, r, pk(sk))$

Equational theory

To model the commitment scheme we need the following two equations:

1. $open(commit(n, r, pk(sk)), r, sk) = n$
2. $commit(n_2, fake(n_1, r, sk, n_2), pk(sk)) = commit(n_1, r, pk(sk))$

Using Maude's Church-Rosser checker we produce their Knuth-Bendix completion:

3. $open(commit(n_1, r, pk(sk)), fake(n_1, r, sk, n_2), sk) = n_2$

Equational theory

To model the commitment scheme we need the following two equations:

1. $open(commit(n, r, pk(sk)), r, sk) = n$
2. $commit(n_2, fake(n_1, r, sk, n_2), pk(sk)) = commit(n_1, r, pk(sk))$

Using Maude's Church-Rosser checker we produce their Knuth-Bendix completion:

3. $open(commit(n_1, r, pk(sk)), fake(n_1, r, sk, n_2), sk) = n_2$

However, this system of equations is still not confluent, to make it so we need to add the following equation:

4. $fake(n_2, fake(n_1, r, sk, n_2), sk, n_3) = fake(n_1, r, sk, n_3)$

Checked Properties

Vote Privacy

- We build two models S_L, S_R using diff terms where the two voters swap candidates.
- Tamarin proves that $S_L \approx_E S_R$
- We adapt the definitions from Delaune et al. 2008 to multiset-rewrite rules

Receipt Freeness

- We substitute rule $V2$ with two rules:
  1. One for the coerced voter, who reveals all his secret information, along with a *fake* or *real* opening of the commitment
  2. One for the colluding voter, who reveals his tracker number

Results

Vote Privacy ✓

- As long as the voters are honest, the attacker cannot distinguish between the two systems

Results

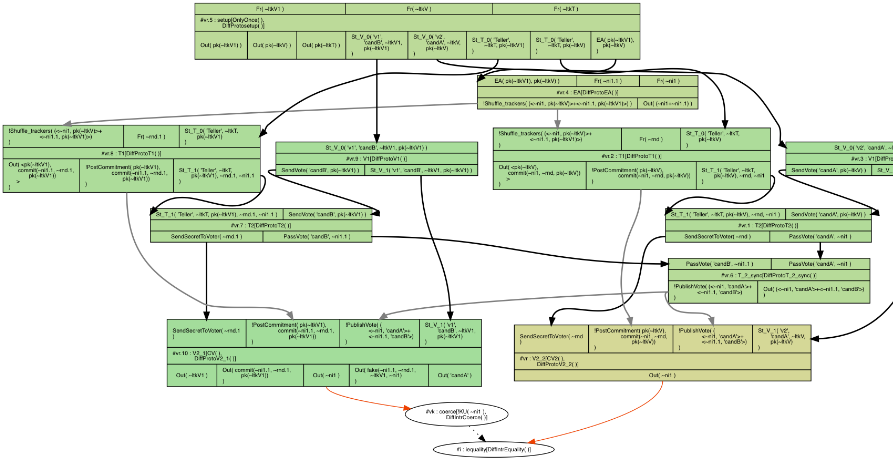
Vote Privacy ✓

- As long as the voters are honest, the attacker cannot distinguish between the two systems

Receipt Freeness ✗

- If the coerced voter hands a fake receipt for the tracker number of the colluding voter, the attacker can find out
- This attack is known (Ryan et. al 2016)

Results



Results

Vote Privacy ✓

- As long as the voters are honest, the attacker cannot distinguish between the two systems

Receipt Freeness ✓

- If the coerced voter hands a fake receipt for the tracker number of the colluding voter, the attacker can find out
- This attack is known (Ryan et. al 2016)
- However if the coerced voter is given *n* fake tracking numbers for each candidate to chose from, then the property holds
- We check this by extending our Tamarin model