Formalising Receipt-freeness

Hugo Jonker

hugo.jonker@uni.lu, h.l.jonker@tue.nl

SaToSS group, University of Luxembourg

FM group, Eindhoven University of Technology

privacy in voting

Receipt-freeness is a particular notion of anonymity in voting.

There are more notions.

- anonymity
- receipt-freeness
- coercion-resistance

No privacy = no free voting

Roughly:

- anonymity

privacy notions

Roughly:

- anonymity
  no observer knows how any voter voted

- receipt-freeness

privacy notions

Roughly:

- anonymity
  no observer knows how any voter voted
- receipt-freeness
  no votebuying
- coercion-resistance

privacy notions

Roughly:

- anonymity
  no observer knows how any voter voted
- receipt-freeness
  no votebuying
- coercion-resistance
  a voter can always fool an observer and still vote freely

Hugo's guide to convincingly selling your vote

1. cast signed vote
2. point to vote in result
3. rich!

Hugo’s guide to convincingly selling your vote

1. cast signed vote
2. point to vote in result
3. rich!

Problem: no signatures in result

1. cast signed vote
2. point to vote in result
3. rich!

Problem: no signatures in result

1. cast encrypted vote
2. point to vote in result, give key
3. rich!

Hugo's guide to convincingly selling your vote

1. cast signed vote
2. point to vote in result
3. rich!

Problem: no signatures in result

1. cast encrypted vote
2. point to vote in result, give key
3. rich!

Problem . . . new guide. . . problem. . .

classical RF

Definition 1 (classical receipt-freeness) *A voting protocol has a receipt iff after execution of the protocol, the voter can provide the intruder with information that proves how she voted.*

A protocol that does not have such a receipt is (classical) receipt-free.

Corollary. Take *extreme* care with voter-supplied randomness!

special channels

Example: using randomness from the voting authority (BT94):

1. Auth provides list of encypted ballots listing all options:
   Ballots $(a_0, b_0), \ldots, (a_n, b_n)$, s.t.
   $\forall i \colon (a_i, b_i) \in \{0, 1\}_{ki} \vee (a_i, b_i) \in \{1, 0\}_{ki}$

2. Send decryptions of a_i to voter over private, untappable channel (commit)

3. Prove that all ballots match ballot 0 (by opening half and linking other half)

4. Voter: send a_0 or b_0 to cast vote of choice.

types of channels

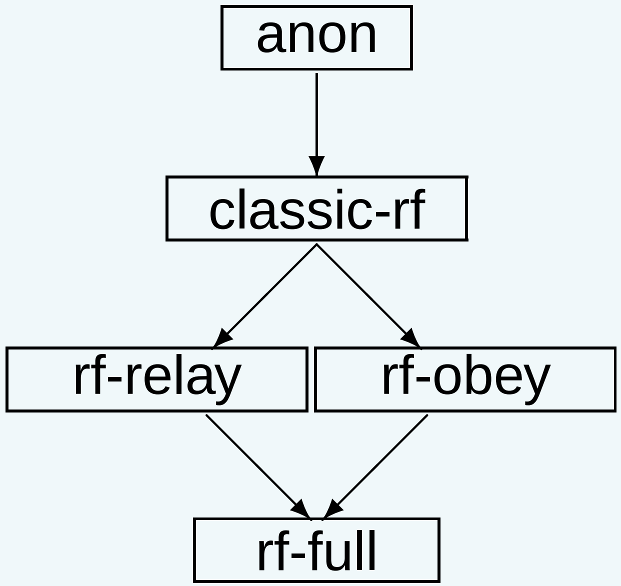
a. public channel

c. untappable channel authority $\rightarrow$ voter

d. untappable channel voter$\rightarrow$ authority

e. untappable channel voter$\leftrightarrow$ authority

privacy attackers



formal model

- voters $\mathcal{V}$, authorities $Auth$, choices $\mathcal{C}$, terms.
- terms are communicated: events.
- events follow eachother: traces.

- parameterize over choice function $\gamma \colon \mathcal{V} \to \mathcal{C}$
- focus on the communication between the parties
  $\Longrightarrow$ different primitives for different channels
- expressed in process algebra (trace semantics)

- idea: measure privacy in anonymity groups

choice groups

Definition 2 (choice groups) *the choice group of voter v in trace $t1$ contains all those candidates, on who v could have voted according to the intruder, who has observed trace $t1$.*

$$cg(v, t1) = \{\gamma_{t2}(v) \mid t2 \in Tr(\mathcal{VS}) \wedge t1 \sim t2\}$$

choice groups

Definition 4 (choice groups) *the choice group of voter v in trace $t1$ contains all those candidates, on who v could have voted according to the intruder, who has observed trace $t1$.*

$$cg(v, t1) = \{\gamma_{t2}(v) \mid t2 \in Tr(\mathcal{VS}) \wedge t1 \sim t2\}$$

Definition 5 (observational equivalence of traces) *Traces t, t' are observationally equivalent with respect to knowledge set K, notation $t \sim t'$, if*

$$\exists \pi : \pi \text{ is a reinterpretation } \wedge t = \pi(t).$$

reinterpretation

Definition 6 (reinterpretation of messages) *(by Garcia et al)*
Let π be a permutation on the set $Terms$ of terms and let K_I be a knowledge set. The map π is said to be a reinterpretation under K_I *if it and its inverse satisfy the following:*

$$\begin{aligned}
\pi(p) &= p && \text{for } p \in \mathcal{C} \cup Nonces \cup Keys \\
\pi((\varphi_1, \varphi_2)) &= (\pi(\varphi_1), \pi(\varphi_2)) && \\
\pi(\{\varphi\}_k) &= \{\pi(\varphi)\}_k && \text{if } K_I \vdash \varphi, k \ \vee \ K_I \vdash \{\varphi\}_k, k^{-1}
\end{aligned}$$

concluding

Learned:

- different attacker models for privacy
- quantify privacy in voting

concluding

Learned:

- different attacker models for privacy
- quantify privacy in voting

Future work:

- reinterpretation of functions
- write thesis

final slide

Thank you for your attention.

Questions?