Proving Security of Voting Systems
A Crash Course

Dr. Ir. Hugo Jonker
SaToSS group, University of Luxembourg
Why?

- why is security needed?
- why do we need an independent proof?
- why formal methods?
security needed

One example: undue influence

Elections must be *fair*!
Nedap: “our voting machines are not computers... They cannot play chess”.
why formal?

Vendor: “This is a very secure product, and should be certified.”

... Chaos Computer Club: “It should not be certified!! It’s insecure!”

We need an unambiguous security proof.
A voting system runs on:

- hardware, running
- software, implementing
- a communication protocol, based on
- cryptosystems, relying on
- mathematical theory.

We focus on the communication protocol, and ignore the other layers.
communication

- public channels

- anonymous channels
  sender remains anonymous.

- untappable channels
  No one but sender and recipient learns anything, not even that a communication occurred.

**Conjecture (from 2000):** without untappable channels or a voting booth, *receipt-freeness* cannot be achieved together with verifiability.
how to prove security

Two approaches:

- Computational model
  Answers of the form: “There is a (non-)negligible chance ...”

- Symbolic model
  Answers of the form: “here is an attack” or “secure”

There are various methods in either approach.
Detailed explanation of one method in this lecture.
generic proof approach

- Option 1:
  1. understand security notion
  2. model system + environment (intruder!)
  3. define security notion as property of system
generic proof approach

- Option 1:
  1. understand security notion
  2. model system + environment (intruder!)
  3. define security notion as property of system

- Option 2:
  1. . . .
  2. . . .
  2b. model “ideal” behaviour
  3. define security notion as relation between these two
notions of privacy

- **vote-privacy:**
  no outside observer can determine how voter $v$ voted.

- **receipt-freeness/coercion-resistance:**
  no observer can determine how $v$ voted, even if $v$ is cooperating with the observer.
The intruder:

- controls the (public) network,
- *perfect cryptography assumption*,
- anonymous channel: intruder cannot determine sender,
- untappable channel: intruder is unaware.

Furthermore: *closed-world assumption*: what is not explicitly stated as true, is false.
Option 1:
1. ✔ understand privacy
2. model system
determine system behaviour
3. determine privacy as a property of system behaviour

Option 2:
1. . . .
2. model system + conspiring voter
3. determine difference in conspiring privacy and previous privacy

There are other ways to determine privacy, this lecture explains only one way.
modelling systems

A voting system:
- consists of a set of agents
- who *communicate*
- *terms*
- containing their *preferred candidate*

So: formalisation of terms, communication $\implies$ system behaviour
Term $\varphi$:

- $v \in \mathcal{V}$, $c \in \mathcal{C}$, $k \in \text{Keys}$, $n \in \text{Nonces}$
- encryption: $\{\varphi'\}_k$
- pairing: $(\varphi_a, \varphi_b)$. 
Term $\varphi$:

- $v \in V$, $c \in C$, $k \in \text{Keys}$, $n \in \text{Nonces}$

- encryption: $\{\varphi'\}_k$

- pairing: $(\varphi_a, \varphi_b)$.

Communication events:

- $va$ sending $\varphi$ to $vb$: $s(va, vb, \varphi)$
- $vb$ receiving $\varphi$ from $va$: $r(va, vb, \varphi)$
Term $\varphi$:
- $v \in V$, $c \in C$, $k \in Keys$, $n \in Nonces$
- encryption: $\{\varphi\}'_k$
- pairing: $(\varphi_a, \varphi_b)$

Communication events:
- $va$ sending $\varphi$ to $vb$: $s(va, vb, \varphi)$
- $vb$ receiving $\varphi$ from $va$: $r(va, vb, \varphi)$
- anonymously: $as(va, vb, \varphi), ar(vb, \varphi)$
- untappable: $uc(va, vb, \varphi)$
System behaviour = list of events. This is called a trace.

Example:

\[
\text{trace } t = s(va, vb, \varphi) \cdot r(va, vb, \varphi) \cdot as(va, vb, \varphi_a) \cdot \ldots
\]
System behaviour

System behaviour = list of events. This is called a trace.

Example:
\[ \text{trace } t = s(va, vb, \varphi) \cdot r(va, vb, \varphi) \cdot as(va, vb, \varphi_a) \cdot \ldots \]

Remarks:
- order may vary (parallel events, choice in executing events)
- anonymous and untappable communications not (completely) observable
System behaviour = list of events. This is called a trace.

Example:
\[ \text{trace } t = s(va, vb, \varphi) \cdot r(va, vb, \varphi) \cdot as(va, vb, \varphi_a) \cdot \ldots \]

Remarks:
- order may vary (parallel events, choice in executing events)
- anonymous and untappable communications not (completely) observable

\[
\begin{align*}
\text{obstr}(\epsilon) &= \epsilon \\
\text{obstr}(\ell \cdot t) &= \begin{cases} 
\text{obstr}(t) & \text{if } \ell = uc(a, a', \varphi) \\
\text{as}(x, \varphi) \cdot \text{obstr}(t) & \text{if } \ell = as(a, x, \varphi) \\
\ell \cdot \text{obstr}(t) & \text{otherwise}
\end{cases}
\end{align*}
\]
How voters vote is given by a *choice function* $\gamma$. For each voter $v \in V$, $\gamma$ returns $v$’s preferred candidate $\gamma(v)$.

**Example.** $V = \{va, vb\}, C = \{c1, c2, c3\}$.

- $\gamma_a(va) = \gamma_a(vb) = c1$.
- $\gamma_b(va) = c1, \gamma_b(vb) = c2$.
- etc.

**Assumption:** The way voters vote (i.e. which $\gamma$ is used) is independent of the voting system.
determining privacy

Privacy question:

Can the intruder tell for a given trace \( t \), if voters voted according to \( \gamma_a \) or according to \( \gamma_b \)?

Let’s try, for \( t \) from \( \forall S \gamma_a \):
determining privacy

Privacy question:

Can the intruder tell for a given trace $t$, if voters voted according to $\gamma_a$ or according to $\gamma_b$?

Let’s try, for $t$ from $\mathcal{VS}^{\gamma_a}$:

- $t = s(va, A, ca) \cdot \ldots \cdot s(vb, A, cb)$? no privacy.
determining privacy

Privacy question:

Can the intruder tell for a given trace $t$, if voters voted according to $\gamma_a$ or according to $\gamma_b$?

Let’s try, for $t$ from $\mathcal{VS}^{\gamma_a}$:

- $t = s(va, A, ca) \cdot \ldots \cdot s(vb, A, cb)$? no privacy.
- $t = s(va, A, \{ca\}_k) \cdot \ldots \cdot s(vb, A, \{cb\}_k)$? privacy? No.
determining privacy

Privacy question:

Can the intruder tell for a given trace $t$, if voters voted according to $\gamma_a$ or according to $\gamma_b$?

Let’s try, for $t$ from $VS^{\gamma_a}$:

- $t = s(va, A, ca) \cdot \ldots \cdot s(vb, A, cb)$? no privacy.
- $t = s(va, A, \{ca\}_k) \cdot \ldots \cdot s(vb, A, \{cb\}_k)$? privacy? No.
- $t = s(va, A, \{ca, n1\}_k) \cdot \ldots \ldots \cdot s(vb, A, \{cb, n2\}_k)$? privacy?
determining privacy

Privacy question:

Can the intruder tell for a given trace \( t \), if voters voted according to \( \gamma_a \) or according to \( \gamma_b \)?

Let’s try, for \( t \) from \( VS^{\gamma_a} \):

- \( t = s(va, A, ca) \cdot \ldots \cdot s(vb, A, cb) \)? no privacy.
- \( t = s(va, A, \{ca\}_k) \cdot \ldots \cdot s(vb, A, \{cb\}_k) \)? privacy? No.
- \( t = s(va, A, \{ca, n1\}_k) \cdot s(va, A, k) \cdot s(vb, A, \{cb, n2\}_k) \)? privacy ?
- \( t = s(va, A, \{ca, n1\}_k) \cdot s(va, A, k) \cdot s(vb, A, \{cb, n2\}_k) \)? no privacy!

Privacy depends on intruder’s knowledge.
determining privacy

Privacy question:

Can the intruder tell for a given trace $t$, if voters voted according to $\gamma_a$ or according to $\gamma_b$?

Let’s try, for $t$ from $VS^{\gamma_a}$:

- $t = s(va, A, ca) \cdot \ldots \cdot s(vb, A, cb)$? no privacy.
- $t = s(va, A, \{ca\}_k) \cdot \ldots \cdot s(vb, A, \{cb\}_k)$? privacy? No.
- $t = s(va, A, \{ca, n1\}_k) \cdot \ldots \cdot s(vb, A, \{cb, n2\}_k)$? privacy?
- $t = s(va, A, \{ca, n1\}_k) \cdot s(va, A, k) \cdot s(vb, A, \{cb, n2\}_k)$!! no privacy!

Privacy depends on intruder’s knowledge.
The intruder can mistake a term $\varphi$ for another term $\varphi'$ as follows:

**Definition 1 (reinterpretation)** Let $\rho$ be a permutation on the set of terms $\text{Terms}$ and let $K_I$ be a knowledge set. The map $\rho$ is a semi-reinterpretation under $K_I$ if it satisfies the following.

\[
\begin{align*}
\rho(p) &= p, \text{ for } p \in C \cup \text{Keys} \cup \mathcal{V} \\
\rho((\varphi_1, \varphi_2)) &= (\rho(\varphi_1), \rho(\varphi_2)) \\
\rho(\{\varphi\}_k) &= \{\rho(\varphi)\}_k, \text{ if } K_I \vdash \varphi, k \land K_I \vdash \{\varphi\}_k, k^{-1}
\end{align*}
\]

Map $\rho$ is a reinterpretation under $K_I$ iff it is a semi-reinterpretation and its inverse $\rho^{-1}$ is a semi-reinterpretation under $\rho(K_I)$. 
Intruder can mistake trace \( t \) for \( t' \), notation \( t \sim t' \), iff he can mistake all the terms in \( t \) for terms in \( t' \), in the same order. Formally:

\[
\exists \rho: \text{obstr}(t') = \rho(\text{obstr}(t)).
\]
Intruder can mistake trace $t$ for $t'$, notation $t \sim t'$, iff he can mistake all the terms in $t$ for terms in $t'$, in the same order. Formally:

$$\exists \rho: \text{obstr}(t') = \rho(\text{obstr}(t)).$$

**Definition 3 (choice indistinguishability)** For voting system $VS$, choice functions $\gamma_a, \gamma_b$ are indistinguishable, $\gamma_a \approx_{VS} \gamma_b$, iff

$$\forall t \in Tr(VS^{\gamma_a}): \exists t' \in Tr(VS^{\gamma_b}): t \sim t' \land$$

$$\forall t \in Tr(VS^{\gamma_b}): \exists t' \in Tr(VS^{\gamma_a}): t \sim t'$$
**Definition 4 (choice group)** Choice group of a given choice function $\gamma$:

$$cg(\mathcal{VS}, \gamma) = \{\gamma' \mid \gamma \simeq_{\mathcal{VS}} \gamma'\}.$$ 

Choice group for a given voter $v$:

$$cg_v(\mathcal{VS}, \gamma) = \{\gamma'(v) \mid \gamma \simeq_{\mathcal{VS}} \gamma'\}.$$ 

Using choice groups, we can define privacy.
Example definitions of privacy

**Definition 5 (privacy I)** Voting system \(\mathcal{VS}\) is private for choice function \(\gamma\) and voter \(v\) iff

\[ cg_v(\mathcal{VS}, \gamma) = \text{set of all candidates who received } \geq 1 \text{ vote}. \]

Or:

**Definition 6 (privacy II)** Voting system \(\mathcal{VS}\) is private for choice function \(\gamma\) and voter \(v\) iff

\[ |cg_v(\mathcal{VS}, \gamma)| > 1. \]

We can test whether a particular voting system complies with a specific privacy definition.
Privacy safeguards:

- voter-secrets (keys)
- untappable channels

A voter may:
Privacy safeguards:
- voter-secrets (keys)
- untappable channels

A voter may:
1. share all her secrets after the elections,
Privacy safeguards:

- voter-secrets (keys)
- untappable channels

A voter may:

1. share all her secrets after the elections,
2. begin by sharing all her secrets,
Privacy safeguards:

- voter-secrets (keys)
- untappable channels

A voter may:

1. share all her secrets after the elections,
2. begin by sharing all her secrets,
3. share everything she receives from an untappable channel,
Privacy safeguards:
- voter-secrets (keys)
- untappable channels

A voter may:
1. share all her secrets after the elections,
2. begin by sharing all her secrets,
3. share everything she receives from an untappable channel,
4. let the intruder determine what to send over an untappable channel.
conspiring voter

Privacy safeguards:
- voter-secrets (keys)
- untappable channels

A voter may:
1. share all her secrets after the elections,
2. begin by sharing all her secrets,
3. share everything she receives from an untappable channel,
4. let the intruder determine what to send over an untappable channel.

Denote this as $cg_v^1(\nuS, \gamma), cg_v^2(\ldots), \ldots$. 
privacy for conspiring voters

classical definition of receipt-freeness:

\[ \forall v, \gamma : \left| cg^1_v(\mathcal{VS}, \gamma) \right| > 1. \]
privacy for conspiring voters

classical definition of receipt-freeness:

\[ \forall v, \gamma : |cg_v^1(\mathcal{VS}, \gamma)| > 1. \]

improved definition: Compare conspiring behaviour with normal behaviour!

Voting system \( \mathcal{VS} \) is \textit{conspiracy-resistant} iff

\[ \forall v \in \mathcal{V}, \gamma \in \mathcal{V} \rightarrow \mathcal{C} : cg_v^i(\mathcal{VS}, \gamma) = cg_v(\mathcal{VS}, \gamma), \]

for \( i \in \{1, 2, 3, 4\} \).
Summary

- Option 1:
  1. understand security notion
  2. model system + environment (intruder!)
  3. define security notion as property of system
     \[\rightarrow\text{ privacy}\]
Summary

- Option 1:
  1. understand security notion
  2. model system + environment (intruder!)
  3. define security notion as property of system  
     \[\Rightarrow\] privacy

- Option 2:
  1. . . .
  2. . . .
  2b. model “ideal” behaviour
  3. define security notion as relation between these two  
     \[\Rightarrow\] privacy for conspiring voter
Thank you for your attention.

Questions?