Electronic Voting: Design, attacks and Formal Verification

Véronique Cortier, CNRS, Loria (Nancy, France)

Joint work with Bruno Blanchet, Vincent Cheval, Alexandre Debant, Pierrick Gaudry, Stéphane Glondu, Lucca Hirschi, Léo Louistisserand, Florian Moser

November 26th, 2025

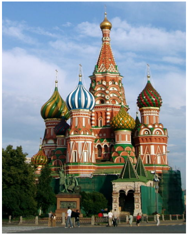
Internet voting is used in various countries

- France: National Assembly, for expats only (2012, 2022, 2024)
- Estonia: local elections (since 2005), national parliamentary elections (since 2007),
  more than 50% of votes cast by Internet in 2023
- Australia: New South Wales state (2021, more than 650 000 votes cast by Internet)
- Switzerland: several trials, a demanding and evolving regulation since 2013
- Canada: local election in Ontario (since 2003) and Nova Scotia (since 2006)

Widely used in non-political election

- professional elections
- associations
- administration councils
- scientific councils

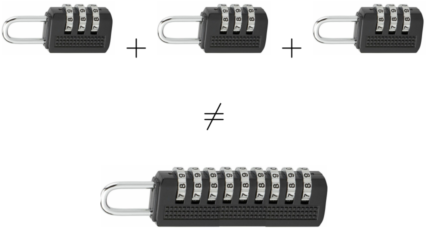
Numerous attacks !



Elections in Moscow, in 2019 [P. Gaudry]

- ballots posted on a blockchain (why?)
- bug bounty program

3 keys of 256 bits $\neq$ 1 key of 768 bits



Numerous attacks !

Swiss context

- open specification, open source code
- call for public scrutiny
- multiple elections in one round

Numerous attacks !

Swiss context

- open specification, open source code
- call for public scrutiny
- multiple elections in one round

Privacy breach with A. Debant and P. Gaudry [RWC'22]

- possibility to (silently) add an extra ballot box, with just Alice' ballot
- a generous bug bounty 

Numerous attacks !

Swiss context

- open specification, open source code
- call for public scrutiny
- multiple elections in one round

Privacy breach with A. Debant and P. Gaudry [RWC'22]

- possibility to (silently) add an extra ballot box, with just Alice' ballot
- a generous bug bounty 

And also

- bad https channel in Australian elections
- complete take-over in overseas US military elections
- PacMan installed on Sequoia Machines AVC Edge
- tampering on voting machines in India
- ...

What is a good voting system?

Confidentiality of the votes

Vote privacy

"No one should know how I voted"

Confidentiality of the votes

Vote privacy

"No one should know how I voted"

Better: Receipt-free / Coercion-resistant

"No one should know how I voted,
even if I am willing to tell my vote! "

Confidentiality of the votes

Vote privacy

"No one should know how I voted"

Better: Receipt-free / Coercion-resistant

"No one should know how I voted,
even if I am willing to tell my vote! "

- vote buying
- coercion

Confidentiality of the votes

Vote privacy

"No one should know how I voted"

Better: Receipt-free / Coercion-resistant

"No one should know how I voted,
even if I am willing to tell my vote! "

- vote buying
- coercion

ebay Silk Road anonymous marketplace

Everlasting privacy: no one should know my vote, even when the cryptographic keys will be eventually broken.

Verifiability

Individual Verifiability: a voter can check that

- cast as intended: their ballot contains their intended vote
- recorded as cast: their ballot is in the ballot box.

Universal Verifiability: everyone can check that

- tallied as recorded: the result corresponds to the ballot box.
- eligibility: ballots have been casted by legitimate voters.

You should verify the election,
not the system.

Verifiability

Individual Verifiability: a voter can check that

- cast as intended: their ballot contains their intended vote
- recorded as cast: their ballot is in the ballot box.

Universal Verifiability: everyone can check that

- tallied as recorded: the result corresponds to the ballot box.
- eligibility: ballots have been casted by legitimate voters.

You should verify the election,
not the system.

Even better: accountability

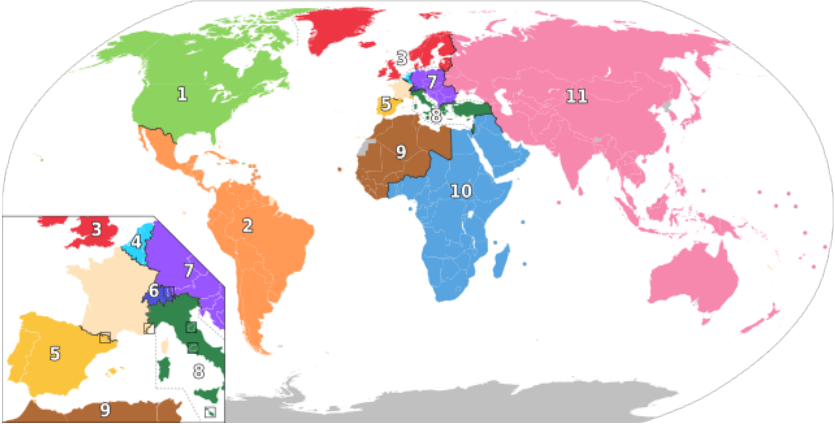
- the system tells whom to blame
- eases dispute resolution

And many more properties

- Availability: servers available at any time
- Accessibility: easy to use, adapted to people with various issues
- ...

2022 French legislative elections

11 circonscriptions (11 deputies), 1.6 M voters.



Crédits: Pierre-Yves Beaudouin / Wikimedia Commons / CC BY-SA 3.0

Context

Voters can vote:

- by postal mail
- at a polling station (at the consulate)
- **by Internet**

Security level required:

Level 3 (the highest) of the CNIL recommendations

This implies verification by **third party tools**.

Objectif de sécurité n° 3-02 : Permettre la transparence de l'urne pour tous les électeurs à partir d'outils tiers.

Building blocks: cryptography

Threshold decryption

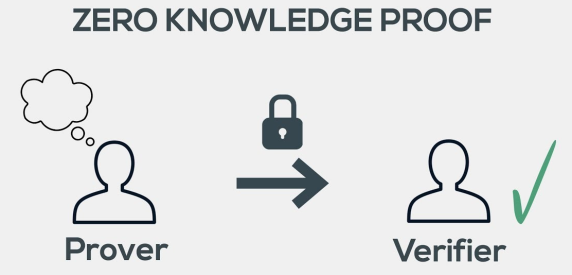
- Each trustee computes her secret key
- The n trustees jointly compute the public key pk
- Decryption with t out of the n keys:
  t out of n trustees suffice to produce decryption shares, that yield the plaintext

Threshold decryption

- Each trustee computes her secret key
- The n trustees jointly compute the public key pk
- Decryption with t out of the n keys:
  t out of n trustees suffice to produce decryption shares, that yield the plaintext

$\rightarrow$ The decryption key is never present on a single computer, neither during the key generation nor the decryption!

Zero-Knowledge proofs



Examples

- Possibility to prove that an encrypted message is either a or b

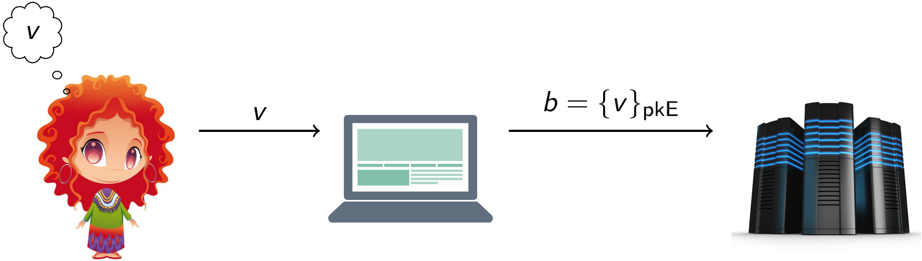
$$\{m\}_k \quad Proof(m = a \text{ or } m = b)$$

- Possibility to prove that the decryption is correct

$$c, m \quad Proof(\mathrm{dec}_k(c) = m)$$

How the FLEP protocol (should) work

Phase 1: vote for $v = 0$ or 1



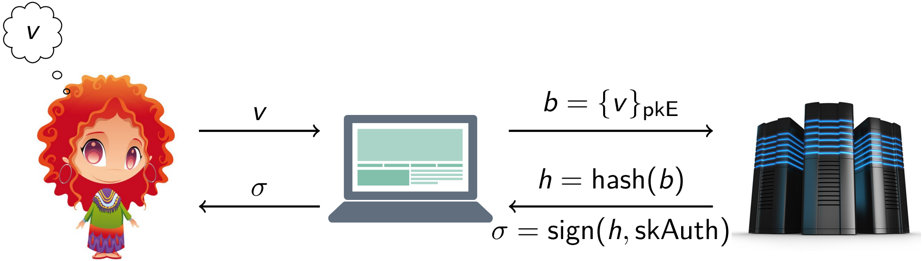
Ballot box

$\{v_a\}_{\mathrm{pkE}}$
$\{v_b\}_{\mathrm{pkE}}$
$\{v_c\}_{\mathrm{pkE}}$

pkE: public key, the private keys are shared among the authorities.

How the FLEP protocol (should) work

Phase 1: vote for $v = 0$ or 1

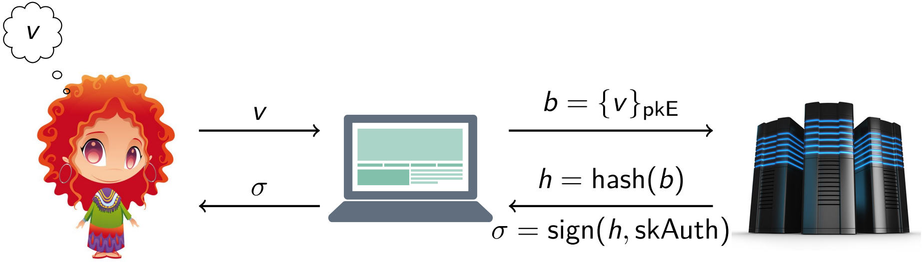


Ballot box
$\{v_a\}_{\mathrm{pkE}}$
$\{v_b\}_{\mathrm{pkE}}$
$\{v_c\}_{\mathrm{pkE}}$
$\{v\}_{\mathrm{pkE}}$

pkE: public key, the private keys are shared among the authorities.

How the FLEP protocol (should) work

Phase 1: vote for $v = 0$ or 1



Ballot box

$\{v_a\}_{\text{pkE}}$
$\{v_b\}_{\text{pkE}}$
$\{v_c\}_{\text{pkE}}$
$\{v\}_{\text{pkE}}$

Phase 2: Tally - homomorphic encryption (El Gamal)

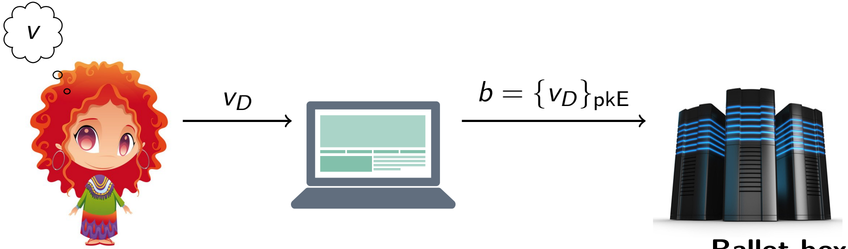
$$\{v_1\}_{\text{pkE}} \times \cdots \times \{v_n\}_{\text{pkE}} = \{v_1 + \cdots + v_n\}_{\text{pkE}}$$

since $g^a \times g^b = g^{a+b}$

$\rightarrow$ Only the final result needs to be decrypted! And proved.

pkE: public key, the private keys are shared among the authorities.

A closer look at ballots - validity

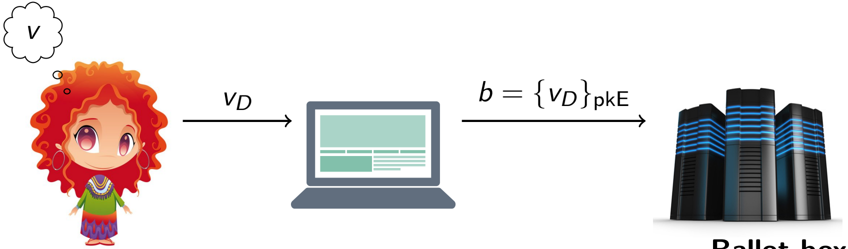


Ballot box

$\{v_a\}_{\text{pkE}}$ $v_A = 0$ or 1
$\{v_b\}_{\text{pkE}}$ $v_B = 0$ or 1
$\{v_c\}_{\text{pkE}}$ $v_C = 0$ or 1
$\{v_D\}_{\text{pkE}}$

Result: $\{v_A + v_B + v_C + v_D + \cdots\}_{\text{pk}(E)}$

A closer look at ballots - validity



Ballot box

$\{v_a\}_{\text{pkE}}$ $v_A = 0$ or 1

$\{v_b\}_{\text{pkE}}$ $v_B = 0$ or 1

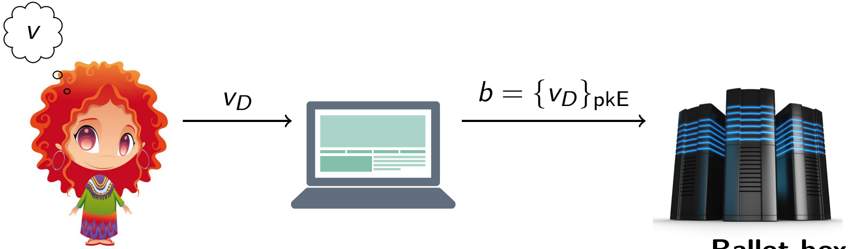
$\{v_c\}_{\text{pkE}}$ $v_C = 0$ or 1

$\{v_D\}_{\text{pkE}}$ $v_D = 100$

Result: $\{v_A + v_B + v_C + 100 + \cdots\}_{\text{pk}(E)}$

A voter could cheat!

A closer look at ballots - validity



Ballot box

$\{v_a\}_{\text{pkE}}$ $v_A = 0$ or 1
$\{v_b\}_{\text{pkE}}$ $v_B = 0$ or 1
$\{v_c\}_{\text{pkE}}$ $v_C = 0$ or 1
$\{v_D\}_{\text{pkE}}$ ~~$v_D = 100$~~

Result: $\{v_A + v_B + v_C + v_D + \cdots\}_{\text{pk}(E)}$

~~A voter could cheat!~~

Use a zero-knowledge proof

$$\{v_D\}_{\text{pk}(E)}, \text{Proof}\{v_D = 0 \text{ or } v_D = 1\}$$

A closer look at ballots - multiple candidates

4 candidates: A, B, C, D

Assume Alice wants to vote for C

candidates	A	B	C	D
vote	0	0	1	0
ballot	$\{0\}_{\text{pkE}}$	$\{0\}_{\text{pkE}}$	$\{1\}_{\text{pkE}}$	$\{0\}_{\text{pkE}}$

$+ \text{Proof}\{v_A = 0 \text{ or } v_A = 1\}, \dots, \text{Proof}\{v_D = 0 \text{ or } v_D = 1\}$

$+ \text{Proof}\{v_A + v_B + v_C + v_D = 1\}$

Voter receipt

Elections législatives 2022 1er tour

Preuve de dépôt du bulletin de vote dans l'urne

Voici la preuve de dépôt de votre bulletin dans l'urne.

Votre bulletin de vote a bien été introduit dans l'urne électronique.

La référence ci-dessous vous permet de contrôler que votre bulletin est bien dans l’urne.

**80011&1&3318f83ea80861c9e6274f049c8df87c2da4fe03e43b7aa46b71
92c0cfc3129c53**

Pour contrôler la référence de votre bulletin : cliquez ici
https://votefae.diplomatie.gouv.fr/pages/verifierEmpreinte

Une fois le dépouillement effectué, vous pouvez vérifier que votre bulletin a bien été pris en compte dans le calcul des résultats, à l’aide d’un outil tiers développé par le CNRS, conformément aux exigences de la CNIL en matière de transparence de l’urne. Pour ce faire, vous devrez renseigner le cachet électronique ci-dessous.

Vous pouvez accédez à l'outil en cliquant ici.

Ce cachet électronique vous permet également de vérifier que votre preuve de vote a bien été produite par le système de vote homologué.

eyJpbmZvU1UiOiI4MDAxMXwxfDFlcmVfQ2lyY29uc2NyaXB0aW9uX2Rlc19GcmFuY2Fpc19kZV9sJ2V0cmFuZ2VyfDE
wOXwzMzE4ZjgzZWE4MDg2MWM5ZTYyNzRmMDQ5YzhkZjg3YzJkYTRmZTAzZTQzYjdhYTQ2YjcxOTJjMGNmYzMxMjljfD
UzIiwic2Nobm9yciI6IjFraWsxaTV2OHQzMGs3NXZhb2htMWhic292aTc1bGE5YWQ2cXBsbmNodXZlajU5c2tub3ElM
TRsdDVkMG9zczEyMGtqdWRtOWUxa2M3MDFsMXRpMWc0bTQza2w4am5qMGhmNmFyNWg4dCIsInB1YmxpY0tleVN1Ijoi
LS0tLS1CRUdJTl9WRVJJRklDQVRJT05fS0VZLS0tLS1cclxuNzdmYTM0ZTQ0YWQxZGI4ZDkxMDg1MmQ4Y2U0ODNkNzc
0YTMyYTRmOTNhMmRlYzRhNjRmNzhmMGFjZmI2NDJjOCUzNjYxNmViNTUxMzY2OWJmZDE2YTdlYTNiZmMzY2Q1NmM3MD
UyMzhlYzk5OTFhNDM0M2QwZTgzOWVjNjM3OTVhXHJcbi0tLS0tRU5EX1ZFUklGSUNBVElPTl9LRVktLS0tLSIsImNsZ
UNhY2hldEJydXQiOiIyOSJ9

Pour contrôler le cachet électronique, cliquez ici

What we did: universal verifiability

Joint work with P. Gaudry and S. Glondu [EVoteID'23]

- Requirement to work on public specifications
- No NDA, responsible disclosure instead

What we did: universal verifiability

Joint work with P. Gaudry and S. Glondu [EVoteID'23]

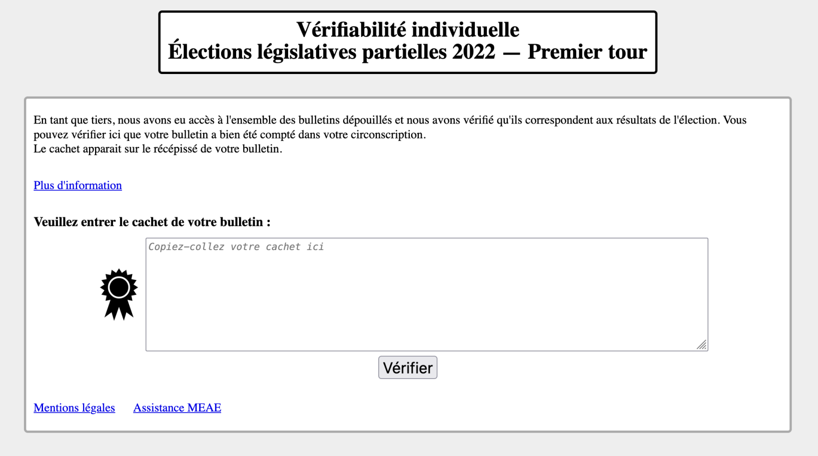
- Requirement to work on public specifications
- No NDA, responsible disclosure instead

After the tally

- we receive the ballot box
- we check the zero-knowledge proofs of correct decryption (and validity of the ballots)
- with our own software, written independently

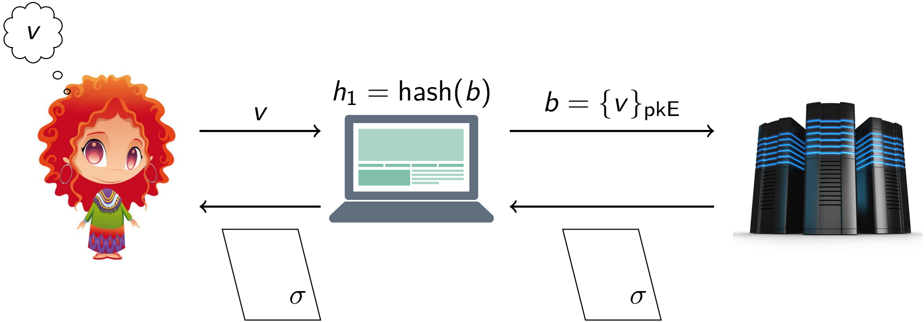
What we did: individual verifiability

- **During the voting phase:** Verification tool for the validity of the server signature;
- **After the tally:** Publication of the list of hashed ballots + verification tool for checking the presence of a hash in this list.



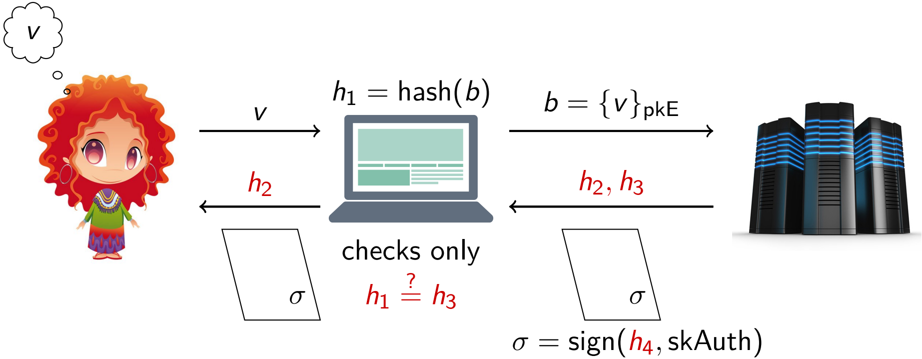
What we missed: several flaws!

A. Debant, L. Hirschi [Usenix'24]



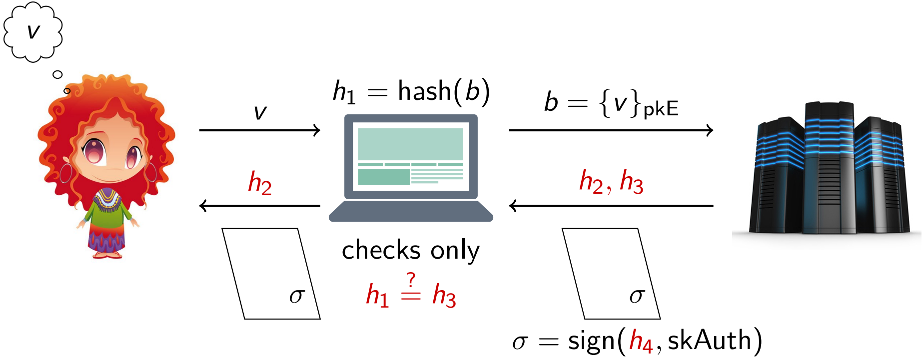
What we missed: several flaws!

A. Debant, L. Hirschi [Usenix'24]



What we missed: several flaws!

A. Debant, L. Hirschi [Usenix'24]



An unsatisfying fix: now both h_1 and h_3 are displayed to the voter for comparison, while the voting client already checks $h_1 = h_3$.
$\rightarrow$ The voter needs to check themself that $h_1 = h_4$, without any instruction.

Formal analysis of e-voting systems

Why a formal analysis of an e-voting system?

⟶ Because formal methods can find attacks before implementations
⟶ Now a current practice for many protocols (TLS, 5G, ...)

Formal analysis of e-voting systems

Why a formal analysis of an e-voting system?

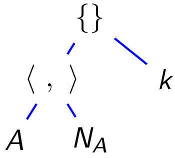
⟶ Because formal methods can find attacks before implementations
⟶ Now a current practice for many protocols (TLS, 5G, ...)

→ Legal requirements in Switzerland to provide **symbolic and cryptographic proofs** of e-voting protocols.

5.1. Examining the cryptographic protocol

5.1.1	Examination criteria: The protocol must meet the security objective according to the trust assumptions in the abstract model in accordance with Section 4. In addition, a cryptographic and a symbolic proof must be provided. The proofs relating to cryptographic basic components may be provided according to generally accepted security assumptions (for example, the "random oracle model", "decisional Diffie-Hellman assumption", "Fiat-Shamir heuristic"). The protocol should be based if possible on existing and proven protocols.

Two main models for security

	Formal approach	Computational approach
Messages		0101000101110101 1101010110101010 0011101011101101 bitstrings
Encryption	terms	algorithm
Adversary	idealized	any polynomial algorithm
Guarantees	some attacks missed	stronger
Proof	often automatic	mostly by hand difficult for complex protocols

Messages

Messages are abstracted by terms.

Agents : $a, b, \ldots$

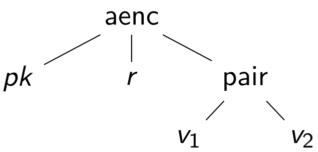
Keys : $k_1, k_2, \ldots$

Ciphertext : $\mathrm{aenc}(pk, r, m)$

Nonces : $n_1, n_2, \ldots$

Concatenation : $\mathrm{pair}(m_1, m_2)$
denoted simply (m_1, m_2) in ProVerif

Example: The encrypted message $\mathrm{aenc}(pk, r, \mathrm{pair}(v_1, v_2))$ is represented by:



Intuition: only the structure of the message is kept.

Model for cryptographic primitives

Projection

$$\pi_1(\mathsf{pair}(x, y)) = x$$

$$\pi_2(\mathsf{pair}(x, y)) = y$$

Asymmetric and symmetric encryption

$$\mathsf{adec}(\mathsf{aenc}(\mathsf{pk}(y), z, x), y) = x$$

$$\mathsf{dec}(\mathsf{enc}(x, y), y) = x$$

Model for cryptographic primitives

Projection

$$\pi_1(\mathsf{pair}(x, y)) = x$$

$$\pi_2(\mathsf{pair}(x, y)) = y$$

Asymmetric and symmetric encryption

$$\mathsf{adec}(\mathsf{aenc}(\mathsf{pk}(y), z, x), y) = x$$

$$\mathsf{dec}(\mathsf{enc}(x, y), y) = x$$

Zero knowledge proof: proof of valid vote

$$\mathsf{aenc}(\mathsf{pk}, r, m), \mathsf{ZKP}(m = 0 \text{ OR } m = 1)$$

$$\begin{aligned} \mathsf{Valid}(\mathsf{ZKP}(\mathsf{aenc}(\mathsf{pk}, r, 0), \mathsf{pk}, r), \mathsf{aenc}(\mathsf{pk}, r, 0), \mathsf{pk}) &= \mathsf{ok} \\ \mathsf{Valid}(\mathsf{ZKP}(\mathsf{aenc}(\mathsf{pk}, r, 1), \mathsf{pk}, r), \mathsf{aenc}(\mathsf{pk}, r, 1), \mathsf{pk}) &= \mathsf{ok} \end{aligned}$$

Syntax for processes

The grammar of processes is as follows:

$$
\begin{aligned}
P, Q, R :=\ & \\
& 0 \\
& \texttt{if } M_1 = M_2 \texttt{ then } P \texttt{ else } Q \\
& \text{let } x = M \text{ in } P \\
& \text{in}(c, x); P \\
& \text{out}(c, N); P \\
& \text{new } n; P \\
& P \mid Q \\
& !P \\
& \text{event} E.P
\end{aligned}
$$

Syntax of ProVerif, a dialect of the applied-pi calculus [AbadiFournet01]

ProVerif: automatic analysis of protocols

Developed by Bruno Blanchet and Vincent Cheval

Performs very well in practice!

- Works on most of existing protocols in the literature
- Is also used on industrial protocols (e.g. TLS, Signal, ...)
- used to pass Swiss requirements on voting
  - Neuchâtel/Scytl protocol [C., Turuani 2018]
  - CHVote protocol [C., Turuani 2019]
  - Swiss Post [Debant,C.,Gaudry, 2022$\rightarrow$now]

ProVerif: automatic analysis of protocols

Developed by Bruno Blanchet and Vincent Cheval

Performs very well in practice!

- Works on most of existing protocols in the literature
- Is also used on industrial protocols (e.g. TLS, Signal, ...)
- used to pass Swiss requirements on voting
  - Neuchâtel/Scytl protocol [C., Turuani 2018]
  - CHVote protocol [C., Turuani 2019]
  - Swiss Post [Debant,C.,Gaudry, 2022→now]

→ ProVerif translates processes in applied pi-calculus into Horn clauses (first-order logic).

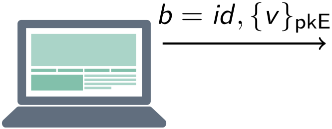
Attacker

Horn clauses perfectly reflects the attacker symbolic manipulations on terms.

$$
\begin{array}{llcll}
\forall x \forall y & I(x), I(y) & \Rightarrow & I(\mathrm{enc}(x,y)) & \text{encryption} \\
\forall x \forall y & I(\mathrm{enc}(x,y)), I(y) & \Rightarrow & I(x) & \text{decryption} \\
\forall x \forall y & I(x), I(y) & \Rightarrow & I(< x, y >) & \text{concatenation} \\
\forall x \forall y & I(< x, y >) & \Rightarrow & I(x) & \text{first projection} \\
\forall x \forall y & I(< x, y >) & \Rightarrow & I(y) & \text{second projection}
\end{array}
$$

Protocol as Horn clauses

let $\mathit{Voter}(\mathrm{pkE}, \mathrm{Vote}, \mathit{id}, \mathrm{cauth}) =$
  new r : bitstring;
  let $b = (\mathit{id}, \mathrm{aenc}(\mathrm{pkE}, r, \mathrm{Vote}))$
  event $\mathit{Voted}(\mathit{id}, \mathrm{Vote}, r)$
  out(cauth, b);
  out(c, b).



Each action of the protocol is translated into logical implications.

$$\forall v \quad I(v) \quad \Rightarrow \quad I(\langle \mathit{id}, \mathrm{aenc}(\mathrm{pkE}, r(v), v\rangle))$$
$$\forall v \quad I(v) \quad \Rightarrow \quad \mathrm{Voted}(\mathit{id}, v, r(v))$$

Security reduces to consistency

secure?

$$
\begin{array}{lrcl}
\forall x \forall y & I(x), I(y) & \Rightarrow & I(< x, y >) \\
\forall x \forall y & I(x), I(y) & \Rightarrow & I(\mathrm{enc}(x, y)) \\
\forall x \forall y & I(\mathrm{enc}(x, y)), I(y) & \Rightarrow & I(x) \\
\forall x \forall y & I(< x, y >) & \Rightarrow & I(x) \\
\forall x \forall y & I(< x, y >) & \Rightarrow & I(y)
\end{array}
$$

$$
\begin{array}{llcl}
\forall v & I(v) & \Rightarrow & I(\langle id, \mathrm{aenc}(\mathrm{pkE}, r(v), v\rangle)) \\
\forall v & I(v) & \Rightarrow & \mathrm{Voted}(id, v, r(v))
\end{array}
$$

Security reduces to consistency

secure?

$\rightsquigarrow$

$$
\begin{array}{lrcl}
 & & & \textsc{Not}\ I(\text{secret}) \\
\forall x \forall y & I(x), I(y) & \Rightarrow & I(< x, y >) \\
\forall x \forall y & I(x), I(y) & \Rightarrow & I(\text{enc}(x, y)) \\
\forall x \forall y & I(\text{enc}(x, y)), I(y) & \Rightarrow & I(x) \\
\forall x \forall y & I(< x, y >) & \Rightarrow & I(x) \\
\forall x \forall y & I(< x, y >) & \Rightarrow & I(y)
\end{array}
$$

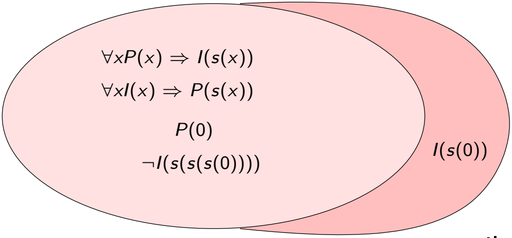
$$
\begin{array}{llcl}
\forall v & I(v) & \Rightarrow & I(\langle id, \text{aenc}(\text{pkE}, r(v), v\rangle)) \\
\forall v & I(v) & \Rightarrow & \text{Voted}(id, v, r(v))
\end{array}
$$

Does not yield a contradiction ?

(i.e. consistent theory ?)

A standard technique: resolution

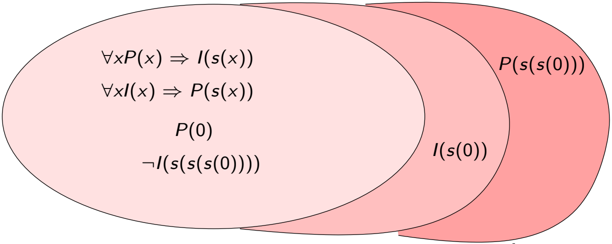
Idea: add logical consequences . . .



... until a contradiction is found.

A standard technique: resolution

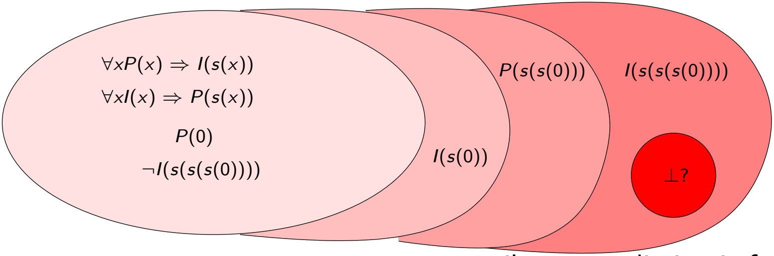
Idea: add logical consequences . . .



... until a contradiction is found.

A standard technique: resolution

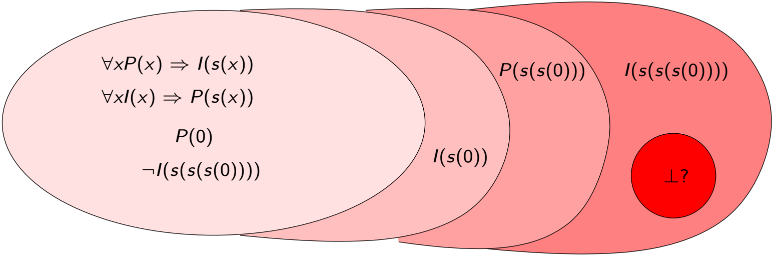
Idea: add logical consequences . . .



... until a contradiction is found.

A standard technique: resolution

Idea: add logical consequences ...



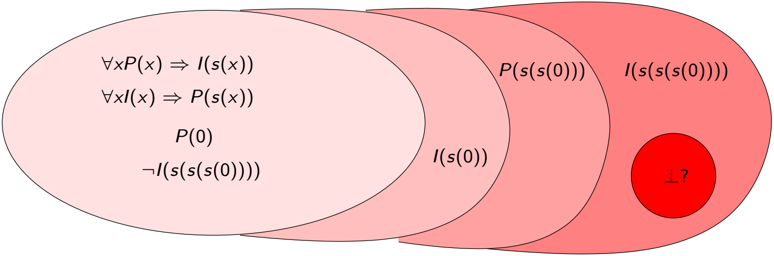
... until a contradiction is found.

Ideally, we need a method (a strategy) which is:

- correct: adds formula that are indeed consequences
- complete: finds a contradiction (if it exists)
- in a finite number of steps

A standard technique: resolution

Idea: add logical consequences . . .



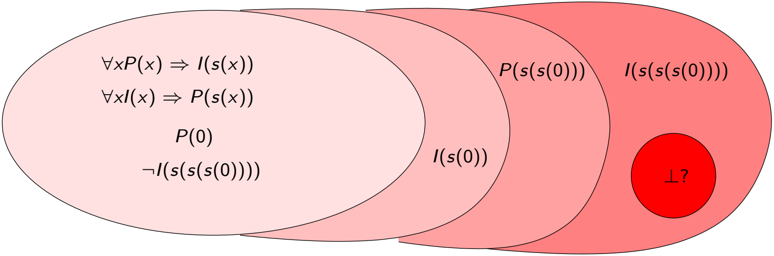
... until a contradiction is found.

Ideally, we need a method (a strategy) which is:

- correct: adds formula that are indeed consequences
- complete: finds a contradiction (if it exists)
- ~~in a finite number of steps~~ undecidable fragment

A standard technique: resolution

Idea: add logical consequences . . .



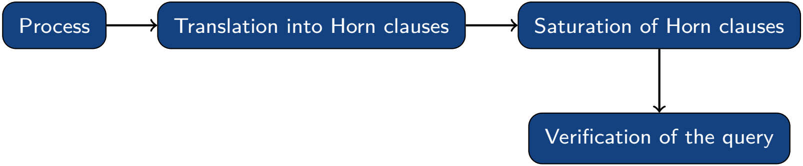
. . . until a contradiction is found.

Ideally, we need a method (a strategy) which is:

- correct: adds formula that are indeed consequences
- ~~complete~~ over-approximations
- ~~in a finite number of steps~~ undecidable fragment

ProVerif

- Implements a correct procedure (that may not terminate or just stop without answer).
- Based on a resolution strategy well adapted to protocols.



Binary resolution

$$\frac{H \Rightarrow C \quad F, H' \Rightarrow C'}{H\sigma, H'\sigma \Rightarrow C'\sigma} \text{ with } \sigma \text{ substitution s.t. } C\sigma = F\sigma$$

- correct
- but adds too many clauses (never terminates)

Binary resolution

$$\frac{H \Rightarrow C \quad F, H' \Rightarrow C'}{H\sigma, H'\sigma \Rightarrow C'\sigma} \text{ with } \sigma \text{ substitution s.t. } C\sigma = F\sigma$$
$$F \neq I(x)$$

- correct
- but adds too many clauses (never terminates)

ProVerif's strategy:

- do not resolve on $I(x)$
- well crafted resolution strategy

Limitations

1. Horn clauses yield over-aproximations
   Example: non uniqueness $\forall v \quad I(v) \Rightarrow \text{Voted}(id, v, r(v))$

Limitations

1. Horn clauses yield over-aproximations
   Example: non uniqueness $\forall v \quad I(v) \Rightarrow \mathrm{Voted}(id, v, r(v))$
   yields
   $\mathrm{Voted}(id, v_1, r(v_1)), \mathrm{Voted}(id, v_2, r(v_2)), \ldots$

Limitations

1. Horn clauses yield over-aproximations
   Example: non uniqueness $\forall v \quad I(v) \Rightarrow \text{Voted}(id, v, r(v))$
   yields
   $\text{Voted}(id, v_1, r(v_1)), \text{Voted}(id, v_2, r(v_2)), \ \ldots$

   Idea: axioms

   $$\text{Voted}(id, v_1, r_1), \text{Voted}(id, v_2, r_2) \Rightarrow v_1 = v_2 \text{ AND } r_1 = r_2$$

Limitations

1. Horn clauses yield over-aproximations
   Example: non uniqueness $\forall v \quad I(v) \Rightarrow \text{Voted}(id, v, r(v))$
   yields
   $\text{Voted}(id, v_1, r(v_1)), \text{Voted}(id, v_2, r(v_2)), \ldots$

   Idea: axioms

   $$\text{Voted}(id, v_1, r_1), \text{Voted}(id, v_2, r_2) \Rightarrow v_1 = v_2 \text{ AND } r_1 = r_2$$

2. Saturation by resolution may still not terminate
   (despite ProVerif's strategy)

Limitations

1. Horn clauses yield over-aproximations
   Example: non uniqueness $\forall v \quad I(v) \Rightarrow \text{Voted}(id, v, r(v))$
   yields
   $\text{Voted}(id, v_1, r(v_1)), \text{Voted}(id, v_2, r(v_2)), \ldots$

   Idea: axioms

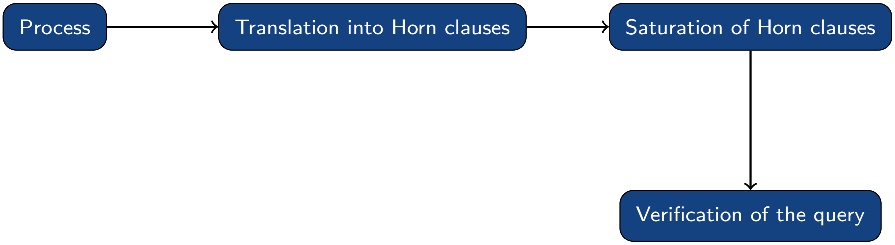
   $$\text{Voted}(id, v_1, r_1), \text{Voted}(id, v_2, r_2) \Rightarrow v_1 = v_2 \text{ AND } r_1 = r_2$$

2. Saturation by resolution may still not terminate
   (despite ProVerif's strategy)

   Idea: lemma as proof helpers

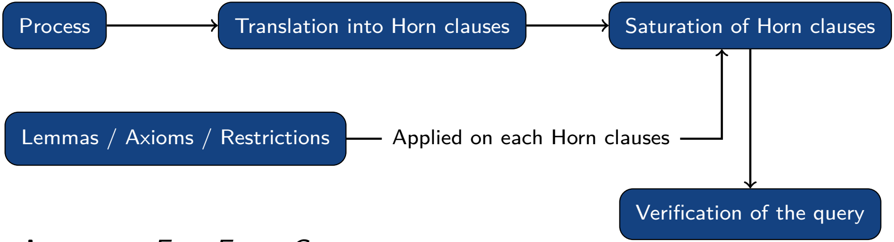
Proverif 2.02: introduction of lemmas

[S&P'22, with B. Blanchet and V. Cheval]



Proverif 2.02: introduction of lemmas

[S&P'22, with B. Blanchet and V. Cheval]



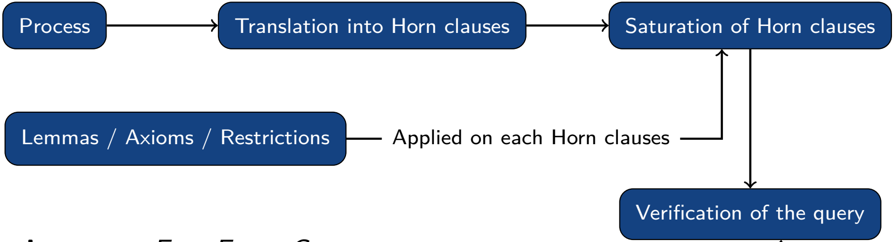
Lemma $F_1 \wedge F_2 \rightarrow G$

Clause $H \Rightarrow C$

If there is a substitution σ s.t. $F_1\sigma, F_2\sigma \subseteq H$ then $H \Rightarrow C$ is replaced by $H \wedge G\sigma \Rightarrow C$

Proverif 2.02: introduction of lemmas

[S&P'22, with B. Blanchet and V. Cheval]



Lemma $F_1 \wedge F_2 \rightarrow G$

Clause $H \Rightarrow C$

If there is a substitution σ s.t. $F_1\sigma, F_2\sigma \subseteq H$ then $H \Rightarrow C$ is replaced by $H \wedge G\sigma \Rightarrow C$

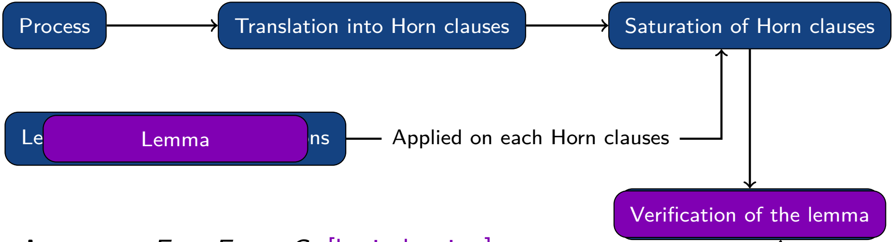
not always sound!

events ✓

attacker facts ✗

Proverif 2.02: introduction of lemmas

[S&P'22, with B. Blanchet and V. Cheval]



Lemma $F_1 \wedge F_2 \rightarrow G$ [by induction]

Clause $H \Rightarrow C$

If there is a substitution σ s.t. $F_1\sigma, F_2\sigma \subseteq H$ then $H \Rightarrow C$ is replaced by $H \wedge G\sigma \Rightarrow C$

not always sound!

events ✓
attacker facts ✗

Even better: lemma by induction

Some challenges

Better formal verification

- decision procedures for larger equational theory classes
- better tools
- formalise security properties, possibly identifying new ones

Better e-voting systems

- more security properties: no vote buying, everlasting privacy, ...
- less trust assumptions (corrupted computers, ...)
- better authentication

Better regulations

- full public specification → should appear in CNIL 2025!
- third party verification
- clear threat models