Improving Automated Symbolic Analysis of Ballot Secrecy for E-voting Protocols: A Method Based on Sufficient Conditions

Euro S&P 2019

Lucca Hirschi & Cas Cremers

June 19th, 2019
Extremely complex setting

- insecure network
- active attacker
- parties running concurrently

Formal methods

- mathematical & exhaustive analysis
- formal guarantees
- automated & mechanised
Symbolic Model

Cryptographic primitives assumed perfect

- primitives modelled as function symbols & equational theory
- e.g. 
  
  \[
  \text{enc}(\cdot, \cdot), \text{dec}(\cdot, \cdot) \quad \& \quad \text{dec}(\text{enc}(m, k), k) = m
  \]

Security protocols

- each party \(\mapsto\) process in a process algebra

Attacker \(\hat{\mathcal{A}} = \text{network}\) (worst case scenario)

- eavesdrop: he learns all protocol outputs
- injections: he chooses all protocol inputs
Symbolic Model

Cryptographic primitives assumed perfect

- primitives modelled as function symbols & equational theory
- e.g. \( \text{enc}(\cdot, \cdot), \text{dec}(\cdot, \cdot) \) & \( \text{dec}(\text{enc}(m, k), k) = m \)

Security protocols

- each party \( \mapsto \) process in a process algebra

Attacker \( \text{\text{red}} \) = network (worst case scenario)

- eavesdrop: he learns all protocol outputs
- injections: he chooses all protocol inputs

Security properties encoded as:

- reachability statements (e.g. for secrecy)
- or behavioral equivalence statements (e.g. for privacy)

Benefit: high level of automation and tool support!
Symbolic Model

Cryptographic primitives assumed perfect

- primitives modelled as function symbols & equational theory
- e.g. \( \text{enc}(\cdot,\cdot), \text{dec}(\cdot,\cdot) \) & \( \text{dec}(\text{enc}(m,k),k) = m \)

Benefit: high level of automation and tool support!
Symbolic Verification of E-Voting Protocols

Remote E-Voting Protocols:
- actually used: Estonia, Australia, Switzerland, many smaller elections
- 2 crucial properties: verifiability (of the election) and privacy (of the votes)
- hard to get right + extremely strong threat model 😈
Remote E-Voting Protocols:
- actually used: Estonia, Australia, Switzerland, many smaller elections
- 2 crucial properties: verifiability (of the election) and privacy (of the votes)
- hard to get right

This Work:
- improve ballot privacy verification technique
- new verification technique based on sufficient conditions
- extends the scope + more efficient

Federal Law!
Remote E-Voting Protocols:

- **actually used**: Estonia, Australia, Switzerland, many smaller elections
- **2 crucial properties**: verifiability (of the election) and privacy (of the votes)
- hard to get right + extremely strong threat model 😈

This Work: Improve ballot privacy verification technique

- new verification technique based on sufficient conditions
- extends the scope + more efficient
Introduction

I State-of-the-Art & Limitations

II Our Approach: Sufficient Conditions for Privacy

III Conclusion
Applied π-Calculus

Model of messages: function symbols & equational theory

Model of protocols: Process algebra

- Process:

\[
P, Q \quad := \quad \text{in}(c, x).P \quad \quad \text{input} \\
\mid \quad \text{out}(c, m).P \quad \quad \text{output} \\
\mid \quad i : P \quad \quad \text{phase (can be executed } \geq \text{ phase } i) \\
\]

Frame (\(\varphi\)): the set of messages revealed to (\(\sigma\)’s knowledge)

Configuration: \(A = (P; \varphi; j)\) (\(P\) multiset of processes, \(j \in \mathbb{N}\))
Applied $\pi$-Calculus

Model of messages: function symbols & equational theory

Model of protocols: Process algebra

**Process:**

\[ P, Q \quad := \quad \text{in}(c, x).P \quad \text{input} \\
\quad \quad \quad \quad \text{out}(c, m).P \quad \text{output} \\
\quad \quad \quad \quad i : P \quad \text{phase} \quad \text{(can be executed } \geq \text{ phase i)} \\
\quad \quad \quad \quad P \mid Q \quad \text{parallel} \\
\quad \quad \quad \quad ! P \quad \text{replication} \\
\quad \quad \quad \quad \text{if Test then } P \text{ else } Q \quad \text{conditional} \\
\quad \quad \quad \quad \text{new } X.P \quad \text{creation of name} \\
\quad \quad \quad \quad 0 \quad \text{null} \]
Applied $\pi$-Calculus

Model of messages: function symbols & equational theory

Model of protocols: Process algebra

- **Process:**
  
  \[
  P, Q := \begin{cases} 
  \text{in}(c, x).P & \text{input} \\
  \text{out}(c, m).P & \text{output} \\
  i : P & \text{phase (can be executed } \geq \text{ phase } i) \\
  P \parallel Q & \text{parallel} \\
  ! P & \text{replication} \\
  \text{if } Test \text{ then } P \text{ else } Q & \text{conditional} \\
  \text{new } X.P & \text{creation of name} \\
  0 & \text{null} 
  \end{cases}
  \]

- **Frame ($\phi$):** the set of messages revealed to 🧐 (🧑‍💻's knowledge)

- **Configuration:** $A = (\mathcal{P}; \phi; j)$ (\mathcal{P} multiset of processes, $j \in \mathbb{N}$)
## E-Voting and Privacy

### E-Voting Protocol (simplified)

- **Roles as processes**: Voter: $V(o,\mathbb{☑})$ and authorities: $A \in \mathcal{R}$
- **Tally** as a function Tally over frames
- **Honest Trace**: a fixed, full, honest execution of \{\(V(o,\mathbb{☑})\) $\cup$ $\mathcal{R}$\}
E-Voting and Privacy

E-Voting Protocol (simplified)

- Roles as processes: **Voter**: $V(\hat{V}, \text{✉})$ and **authorities**: $A \in \mathcal{R}$
- **Tally** as a function $\text{Tally}$ over frames
- **Honest Trace**: a fixed, full, honest execution of $\{V(\hat{V}, \checkmark)\} \cup \mathcal{R}$

Ballot Privacy (simplified)

\[
V(\hat{V}, \checkmark) \parallel V(\hat{V}, \times) \parallel !\mathcal{A} \approx V(\hat{V}, \times) \parallel V(\hat{V}, \checkmark) \parallel !\mathcal{A}
\]

Where $\approx$ is a behavioral equivalence: \hat{V} cannot tell both sides apart.

“\hat{V} cannot establish meaningful link between a voter and his vote”
E-Voting and Privacy

E-Voting Protocol (simplified)

- Roles as processes: Voter: \( V(\bullet, \checkmark) \) and authorities: \( A \in \mathcal{R} \)
- Tally as a function Tally over frames
- Honest Trace: a fixed, full, honest execution of \( \{V(\bullet, \checkmark)\} \cup \mathcal{R} \)

Ballot Privacy (simplified)

\[ V(\bullet, \checkmark) \mid V(\bullet, \times) \mid !A \approx V(\bullet, \times) \mid V(\bullet, \checkmark) \mid !A \]

Where \( \approx \) is a behavioral equivalence: \( \hat{\bullet} \) cannot tell both sides apart.

Trivial Example:

\[ V(\bullet, \checkmark) := 1 : \text{out}(c, \bullet).\text{out}(c, \hat{\bullet}) \]
E-Voting and Privacy

E-Voting Protocol (simplified)

- Roles as processes: **Voter**: \( V(\square, \text{☑}) \) and **authorities**: \( A \in \mathcal{R} \)
- **Tally** as a function \( \text{Tally} \) over frames
- **Honest Trace**: a fixed, full, honest execution of \( \{ V(\square, \text{☑}) \} \cup \mathcal{R} \)

Ballot Privacy (simplified)

\[
V(\square, \text{☑}) \mid V(\square, \text{✗}) \mid !A \approx V(\square, \text{✗}) \mid V(\square, \text{☑}) \mid !A
\]

Where \( \approx \) is a behavioral equivalence: \( \square \) cannot tell both sides apart.

Trivial Example: 
\( V(\square, \text{☒}) := 1 : \text{out}(c, \square) \cdot \text{out}(c, \text{☒}) \) attack \( \square \)!

In \( V(\square, \text{☑}) \mid V(\square, \text{✗}) \), \( \square \) can “block” \( \square \) and observes \( \square \)’s \( \text{☒} \): \( \text{☑} \neq \text{✗} \).
E-Voting and Privacy

E-Voting Protocol (simplified)

- Roles as processes: **Voter**: $V(\bullet, \square)$ and **authorities**: $A \in \mathcal{R}$
- **Tally** as a function $\text{Tally}$ over frames
- **Honest Trace**: a fixed, full, honest execution of $\{V(\bullet, \checkmark)\} \cup \mathcal{R}$

Ballot Privacy (simplified)

$$V(\bullet, \checkmark) \parallel V(\bullet, \times) \parallel !A \approx V(\bullet, \times) \parallel V(\bullet, \checkmark) \parallel !A$$

Where $\approx$ is a behavioral equivalence: $\bullet$ cannot tell both sides apart.

Trivial Example:

$$V(\bullet, \square) := 1: \text{out}(c, \bullet). \ 2: \text{out}(c, \square) \quad \text{secure 😊}$$

$\leadsto \bullet$ has to let both $\bullet$ and $\bullet$ reach phase 2 before getting any $\square$
Problem

State-of-the-art: $\approx$ approximated by “diff-equivalence” (when $\infty$ sessions)

Ballot privacy: $V(\hat{a},\checkmark) \mid V(\hat{a},\times) \mid \neg A \approx V(\hat{a},\times) \mid V(\hat{a},\checkmark) \mid \neg A$
Problem

State-of-the-art: \( \approx \) approximated by “diff-equivalence” (when \( \infty \) sessions)

Ballot privacy: \( V(\hat{\mathcal{P}}, \text{diff}[\checkmark, \times]) \mid V(\hat{\mathcal{P}}, \text{diff}[\times, \checkmark]) \mid \neg \mathcal{A} \)
Problem

State-of-the-art: $\approx$ approximated by “diff-equivalence” (when $\infty$ sessions)

Ballot privacy: $V(\hat{\cdot}, \text{diff}[\checkmark, \times]) \mid V(\hat{\cdot}, \text{diff}[\times, \checkmark]) \mid !A$

$\text{diff-equivalence} = \approx \text{for } \hat{\cdot} \text{ who knows internal structure of processes}$

Implications:

- $\hat{\cdot}$ knows when actions are triggered by the same process/agent

Structural links given to $\hat{\cdot}$ vs. ballot privacy=absence of certain links:

$\leadsto$ systematic false attacks on ballot secrecy

$\leadsto$ ad hoc work-arounds with limited applicability e.g. swaps of processes
Our hybrid approach: privacy via sufficient conditions

Methodology:

▸ focus on some class of protocols and some privacy goal
▸ identify conditions (inspired by generic classes of attacks)
▸ that are sufficient (soundness),
▸ fundamentally simpler and easier to check (checkability), and
▸ met by (secure) protocols (tightness)
Our hybrid approach: privacy via sufficient conditions

Methodology:

▸ focus on some class of protocols and some privacy goal
▸ identify conditions (inspired by generic classes of attacks)
▸ that are sufficient (soundness),
▸ fundamentally simpler and easier to check (checkability), and
▸ met by (secure) protocols (tightness)

Goal: More precise & efficient verification techniques + extends the scope.

First developed for untraceability:

Introduction

I State-of-the-Art & Limitations

II Our Approach: Sufficient Conditions for Privacy

III Conclusion
Leaking Status

Take for instance: \[ V(\text{Voter}, \text{Mask}) = \text{new } n.1 : \text{out(\text{Voter})}.P.\text{out(Mask)}.\text{out(n)} \]

\[
\begin{array}{|c|c|c|}
\hline
V = 1: & \text{Out(H)} & \text{In(y)} & \text{Out(u)} \\
\hline
& \text{Out(V)} & \text{Out(n)} \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|}
\hline
A = 1: & \text{In}(x) & \text{Out(t)} & \text{In(z)} \\
\hline
\end{array}
\]
Leaking Status

Take for instance:

\[
V(\begin{array}{|c|}
\hline
\text{id} \\
\hline
\end{array}, \begin{array}{|c|}
\hline
\text{v} \\
\hline
\end{array}) = \text{new } n.1 : \text{out}(\begin{array}{|c|}
\hline
\text{id} \\
\hline
\end{array}) . P . \text{out}(\begin{array}{|c|}
\hline
\text{v} \\
\hline
\end{array}) . \text{out}(n)
\]

\[V = 1:\]

- \text{Out(id)}
- \text{In(y)}
- \text{Out(u)}

- \text{Out(v)}
- \text{Out(n)}

\[A = 1:\]

- \text{In(x)}
- \text{Out(t)}
- \text{In(z)}

At most 1 type of leak in a single phase; phase leaking status id-leaking phases unlinkable to vote \(\land\) vote-leaking phases unlinkable to id

Similarly: name has at most 1 type of leak; name leaking status
Leaking Status

Take for instance: $V(\bullet, \square) = \text{new } n. 1: \text{out}(\bullet). P. 2: \text{out}(\square) . \text{out}(n)$

$V = 1$:  
- Out(\text{id})
- In(y)
- Out(u)

$A = 1$:  
- In(x)
- Out(t)
- In(z)

$V = 2$:  
- Out(V)
- Out(n)
Leaking Status

Take for instance: $V(\text{id}, \text{vote}) = \text{new } n.1 : \text{out(\text{id})}.P. 2 : \text{out(\text{vote})}.\text{out}(n)$

At most 1 type of leak in a single phase $\leadsto$ phase leaking status

- id-leaking phases unlinkable to vote
- vote-leaking phases unlinkable to id
Leaking Status

Take for instance: $V(\text{id}, \text{vote}) = \text{new } n.1 : \text{out}(\text{id}).P. 2 : \text{out}(\text{vote}).\text{out}(n)$

- $V = 1$: 
  - Out(id)
  - In(y)
  - Out(u)

- $A = 1$: 
  - In(x)
  - Out(t)
  - In(z)

id-leaking phase ▶ vote-leaking phase

- At most 1 type of leak in a single phase ◄ phase leaking status
  - id-leaking phases unlinkable to vote
  - vote-leaking phases unlinkable to id

- Similarly: name has at most 1 type of leak ◄ name leaking status
1: Dishonest Condition

Idea: if a deviation from the honest execution at phase $i$ has some impact at phase $j > i$, it may link phases $i$ and $j$.

- e.g. taint credential at phase 1 and observe it at phase 2

Dishonest Condition (Informal)
For any execution, if a voter process $V$ at phase $j$ is still present at the end, then it followed the honest trace up to $j - 1$.

- Prevent a class of attacks
- Allow us to focus on less executions (those that meet the condition)

\[ V = 1: \quad \text{Out}(\id) \quad \text{Out}(u) \]
\[ A = 1: \quad \text{In}(x) \quad \text{Out}(t) \quad \text{In}(z) \]
\[ 2: \quad \text{Out}(\top) \quad \text{Out}(n) \]

past = honest execution
1: Dishonest Condition

Idea: if a deviation from the honest execution at phase $i$ has some impact at phase $j > i \implies \exists$ may link phases $i$ and $j$.

- e.g. taint credential at phase 1 and observe it at phase 2

Dishonest Condition (Informal)

For any execution, if a voter process $V$ at phase $j$ is still present at the end, then it followed the honest trace up to $j - 1$.

- Prevent a class of attacks
- Allow us to focus on less executions (those that meet the condition)

\[
R^d(\text{id}_A, n^v_1) = \{ 1: \text{Out}(A) \quad \text{In}(y) \quad \text{Out}(u), \\
1: \text{In}(x) \quad \text{Out}(t) \quad \text{In}(z) \}
\]

\[
R^v(\text{id}_A, n^v_1) = \{ 2: \text{Out}(n_1) \}
\]

less structural links with “standalone phase-processes” 😊
We would like to check the absence of relation for all phase-processes.

(less structural links now 😊)

diff[nᵥ, nₓ] in id-leaking phase-processes

Defined as the diff-equivalence of:

\[ \mathcal{B} = \{ \mathcal{R}^{id}(n^{id}_n, \text{diff}[n^v_n, n^x_n]), \]

\[ \mathcal{R}^{id}(n^{id}_n, \text{diff}[n^y_n, n^x_n]) \}

∪ !\mathcal{R}
2: Relation Condition

We would like to check the absence of $\bullet - \square$ relation for all phase-processes.

(less structural links now 😊)

diff[$n^v$, $n^x$] in id-leaking phase-processes

diff[$n^{id}$, $n^{id}$] in vote-leaking phase-processes

Defined as the diff-equivalence of:

$$\mathcal{B} = \{ \mathcal{R}^{id}(n^{id}, \text{diff}[n^v, n^x]), \quad \mathcal{R}^v(\text{diff}[n^{id}, n^{id}], n^v),$$

$$\mathcal{R}^{id}(n^{id}, \text{diff}[n^x, n^v]), \quad \mathcal{R}^v(\text{diff}[n^{id}, n^{id}], n^x)$$

$$\} \cup !\mathcal{R}$$
We would like to check the absence of \(\rightarrow\) relation for all phase-processes.

(less structural links now 😊)

\[
\text{diff}[n^\text{v}, n^\text{x}] \text{ in id-leaking phase-processes}
\]

\[
\text{diff}[n^\text{id}, n^\text{id}] \text{ in vote-leaking phase-processes}
\]

Defined as the diff-equivalence of:

\[
\mathcal{B} = \{ \mathcal{R}^\text{id}(n^\text{id}, \text{diff}[n^\text{v}, n^\text{x}]), \quad \mathcal{R}^\text{v}(\text{diff}[n^\text{id}, n^\text{id}], n^\text{v}), \\
\quad \mathcal{R}^\text{id}(n^\text{id}, \text{diff}[n^\text{x}, n^\text{v}]), \quad \mathcal{R}^\text{v}(\text{diff}[n^\text{id}, n^\text{id}], n^\text{x}) \}
\]

\(\uplus \lnot \mathcal{R}\)

**Relation Condition (Informal)**

The Honest Relations Condition is satisfied if \(\mathcal{B}\) is diff-equivalent.
Our Results

**Theorem (soundness)**

For any $E = (V(\text{Dishonest}, \text{Relation}), \mathcal{R}, \text{Tally})$, if the Dishonest, Relation, and Tally conditions hold then $E$ satisfies ballot secrecy.

(Tally condition omitted)

- We provide an algorithm for computing models checking the conditions and heuristics to find leaking status (checkability) (tool is FW)
- We verify some case studies + benchmarks (tightness):

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Ballot Secrecy</th>
<th>Our verif. time</th>
<th>Previous state of the art</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOO</td>
<td>✔️</td>
<td>0.04</td>
<td>0.26</td>
</tr>
<tr>
<td>Lee 1</td>
<td>✔️</td>
<td>0.04</td>
<td>46</td>
</tr>
<tr>
<td>Lee 2</td>
<td>✔️</td>
<td>0.05</td>
<td>†</td>
</tr>
<tr>
<td>Lee 3</td>
<td>✔️</td>
<td>0.01</td>
<td>†</td>
</tr>
<tr>
<td>Lee 4</td>
<td>☹️</td>
<td>6.64</td>
<td>169.94</td>
</tr>
<tr>
<td>JCJ</td>
<td>✔️</td>
<td>18.79</td>
<td>✗</td>
</tr>
<tr>
<td>Belenios</td>
<td>✔️</td>
<td>0.02</td>
<td>✗</td>
</tr>
</tbody>
</table>

曜: false attack  †: non-termination (>45h)
Introduction

I Privacy via Sufficient Conditions

II Application to E-Voting

III Conclusion
Conclusion

Summary

- Three tight, sufficient conditions for ballot privacy
- Expands the class of protocols and threat models that can be verified
- More efficient verification

Future Work

- Extend our result with more precise Tally
- Combine with the new BPRIV privacy definition [S&P’15, Euro S&P’19]
- Provide a tool with ProVerif/Tamarin as back-end
- Reuse methodology for other contexts/privacy properties

lucca.hirschi@inria.fr
Backup Slides
Symbolic Model

Big Picture

Protocol's specification $\xrightarrow{X} \text{Protocol's model}$

$P \models \text{in}(x).
\text{new } Y.
\text{out(\text{enc}((x, Y, k)))}$

$P \models \ldots$

$\approx$ between transition systems

Privacy goal $\xleftarrow{\text{e.g. cannot track}} \approx$ between scenarios

e.g. $\ldots$, $\approx \ldots$, $\approx \ldots$
Two Approaches for Verifying $\approx$ Automatically

Decision for $< \infty$ sessions

- **bound** the number of sessions
- **symbolic** semantics
  - $\leadsto$ finite description of $\approx$
- **exhaustive exploration** of symbolic executions
- Tools: Apte, Akiss, Spec

Semi-decision for $\infty$ sessions

- over-approximations of $\approx$ & semantics
- **strong** form of $\approx$ (i.e. diff-equivalence)
- Tools: ProVerif, Tamarin, Maude-NPA
Limitation of Semi-decision Procedures

- Serious Precision Issue (privacy)
- \( \leadsto \) systematic false attacks for e.g. unlinkability, vote-privacy (e-Passport, RFID protocols, 4G, e-voting ...)

Semi-decision for \( \infty \) sessions

- over-approximations of semantics
- strong form of \( \approx \) (i.e. diff-equivalence)

Tools: ProVerif, Tamarin, Maude-NPA
Applied $\pi$-Calculus

Model of messages: Term algebra

- Function symbols
- Equational theory $=_E +$ computation relation $\downarrow$

Model of protocols: Process calculus

- Process: $P, Q := 0$ null
  - in$(c, x).P$ input
  - out$(c, m).P$ output
  - let $x = v$ then$P$ else $Q$ conditional
  - $P | Q$ parallel
  - $! P$ replication
  - new $n.P$ creation of name
  - $i : P$ weak phase

- Frame $(\phi)$: the set of messages revealed to $(\mathcal{M}'$s knowledge)

$$\phi = \{ w_1 \mapsto \text{enc}(m, k), w_2 \mapsto k \}$$

- Configuration: $A = \langle P; \phi; j \rangle$
**Applied-$\pi$ - Semantics**

- **Recipes**: terms built using handles

  \[
  \begin{align*}
  R &= \text{dec}(w_1, w_2) \\
  R\phi &=_{E} m \\
  \end{align*}
  \]

  for \( \phi = \{ w_1 \mapsto \text{enc}(m, k), w_2 \mapsto k \} \)

  “How ♻ builds messages from its knowledge”
Applied-$\pi$ - Semantics

- **Recipes**: terms built using handles

  \[ R = \text{dec}(w_1, w_2) \quad \text{for} \quad \phi = \{w_1 \mapsto \text{enc}(m, k), w_2 \mapsto k\} \]

  "How 🐷 builds messages from its knowledge"

- Protocol’s output:

  \[ (\{i : \text{out}(c, u).P\} \cup \mathcal{P}; \phi; i) \xrightarrow{\text{out}(c, w)} (\{i : P\} \cup \mathcal{P}; \phi \cup \{w \mapsto u\}; i) \quad \text{if } w \text{ fresh} \]

- Protocol’s input:

  \[ (\{i : \text{in}(c, x).P\} \cup \mathcal{P}; \phi; i) \xrightarrow{\text{in}(c, R)} (\{i : P\{x \mapsto R\phi\}\} \cup \mathcal{P}; \phi; i) \]

- + expected rules for conditional (modulo $=E$) & others

  🐼 controls all the network
Applied-$\pi$ - Trace Equivalence

Static Equivalence (intuitively)

$\Phi \sim \Psi$ when

- $\text{dom}(\Phi) = \text{dom}(\Psi)$ and
- for all tests, it holds on $\Phi \iff$ it holds on $\Psi$ (modulo $=_E$)

Trace Equivalence

$A \approx B$: for any $A \xrightarrow{t} A'$ there exists $B \xrightarrow{t'} B'$ such $\Phi(A') \sim \Phi(B')$ and $\text{obs}(t) = \text{obs}(t')$

(and the converse).
Privacy

Unlinkability

\[ \mathcal{M} := !\text{new Id. } !\text{new Sess.}(P_{\text{\textbullet}} | P_{\text{\textbullet}}) \approx? !\text{new Id. } \text{new Sess.}(P_{\text{\textbullet}} | P_{\text{\textbullet}}) \]

\(\triangleright\)
cannot establish meaningful link between two interactions (with same Id)

Anonymity

\[ \mathcal{M} \mid !\text{new Sess.}(P_{\text{\textbullet}}(\text{Id}_0) | P_{\text{\textbullet}}(\text{Id}_0)) \approx? \mathcal{M} \]

\(\triangleright\)
cannot establish meaningful link between an interaction and identity \(\text{Id}_0\)

Ballot Secrecy

\[ V(\text{\textbullet, \checkmark}) \mid V(\text{\textbullet, \times}) \mid !\mathcal{A} \approx? V(\text{\textbullet, \times}) \mid V(\text{\textbullet, \checkmark}) \mid !\mathcal{A} \]

\(\triangleright\)
cannot establish meaningful link between a voter and his vote
Goal: Analyzing Ballot Secrecy

Often, only the core voting protocol is analyzed.

We would like to take into account important aspects such as:

▸ registration
▸ credential delivery
▸ authentication
▸ voting
▸ tallying

We would like to:

▸ compare different threat models (no security if everything is compromised)
▸ identify minimal honesty assumptions

Lucca Hirschi & Cas Cremers
Improving Automated Symbolic Analysis of Ballot Secrecy for E-voting Protocols
Goal: Analyzing Ballot Secrecy

Often, only the core voting protocol is analyzed.

Voter $Id, v, cred$

Registrar $sk_R, cred$

Tallier $sk_T$

$ZK(aenc(cred, r^1_v), pk_T); aenc(v, r^2_v, pk_T); [cred, v, r^1_v, r^2_v])$

$\text{sign}(aenc(cred, r^1_R), pk_T, sk_R)$

PET + MIX + Open
Goal: Analyzing Ballot Secrecy

Often, only the core voting protocol is analyzed.

We would like to take into account important aspects such as:

- registration, credential delivery
- authentication
- voting
- tallying

We would like to:

- compare different threat models (no security if everything is compromised)
- identify minimal honesty assumptions
Verifying Ballot Secrecy

\[ V(\text{ }//, √) \mid V(\text{ }//, ×) \mid !A \approx? V(\text{ }//, ×) \mid V(\text{ }//, √) \mid !A \]

**Diff-equivalence yields false attacks**

Take: \[ V(\text{ }//, ☵) = 1 : \text{out}(c, \text{■}) \]. \[ 2 : \text{out}(c, ☵) \]

With diff-equivalence, \[ ☵ \] can link all actions from \[ §\ ] (resp. \[ ☼\ ])

\[ → \text{attacker can link } \text{■} \text{ and } ☵ \]
State-of-the-Art

Weakening diff-equivalence (improving the tool):

- **Swapping approach** – Idea:[DRS’08], Proof+ProVerif:[BB’16], Tamarin:[DDKS’17]: allows to change biprocess pairing at sync. barriers

Hybrid approaches:

- type system [CGLM’17]

- small attack property [ACK’16]
State-of-the-Art

Weakening diff-equivalence (improving the tool):

- **Swapping approach** – Idea:[DRS’08], Proof+ProVerif:[BB’16], Tamarin:[DDKS’17]: allows to change biprocess pairing at sync. barriers

Limitations:

- no swap/phase under replication
- no honest authority present in ≠ phases
- no threat model with no dishonest voters
- introduction of new internal communication
- false attacks in presence of fresh data going through phases

Hybrid approaches:

- type system [CGLM’17]
- small attack property [ACK’16]
State-of-the-Art

Weakening diff-equivalence (improving the tool):

- **Swapping approach** – Idea:[DRS’08], Proof+ProVerif:[BB’16], Tamarin:[DDKS’17]: allows to change biprocess pairing at sync. barriers

  **Limitations:**
  - no swap/phase under replication
  - no honest authority present in ≠ phases
  - no threat model with no dishonest voters
  - introduction of new internal communication
  - false attacks in presence of fresh data going through phases \((1 : \text{new } n.2 : \text{out}(c, (v, n)))\)

Hybrid approaches:

- type system [CGLM’17] but pairing is as rigid as diff-equivalence, standard primitives only
- small attack property [ACK’16] but only 1 phase, performance issues

In practice, interesting threat models and modeling of e.g. Lee, JCJ, Belenios are out of the scope
Our contribution – Big Picture

We develop a privacy via sufficient conditions approach for ballot secrecy and a large class of e-voting protocols (soundness, checkability, tightness).

We apply our technique on FOO, Lee, JCJ and Belenios (with registration):

▸ false attacks using previous techniques (e.g. JCJ, Belenios)
▸ much better performance (e.g. \( \times 10^2 \), termination for LEE)
Our contribution – Big Picture

We develop a privacy via sufficient conditions approach for ballot secrecy and a large class of e-voting protocols (soundness, checkability, tightness).

We apply our technique on FOO, Lee, JCJ and Belenios (with registration):

▸ false attacks using previous techniques (e.g. JCJ, Belenios)
▸ much better performance (e.g. $\times10^2$, termination for LEE)

Main Limitation:

▸ Tallier is too unrealistic: no revote policy, homomorphic tallying
Class of e-voting protocols

(Honest) Roles:

- **Voter:**  \( V(\text{ }, \text{ }) = i : \text{new } \vec{n}.V' \text{ such that } V' \text{ has no !,| or new} \\
- \( A \in \mathcal{R} \) _authority_ session, same format _+ (?) voters_ \\
- Some role \( A_c \in \mathcal{R} \) is the _bulletin box_ and \( A_b \ni \text{out}(c_b, t) \) _“stores in BB”_

Tally:

- Made of a public term \( \Psi_b \) (correct form?) and private term Extract (check validity and extract vote) \\
- ”Tally” =!i_f : \text{in}(c, x).let \( (_, v) = (\Psi_b[x], \text{Extract}[x]) \) in \text{out}(c, v)

Honest Trace: (symbolic) trace th s.t. \((\mathcal{R} \cup \{V(\text{ }, \checkmark)\}; \phi_0; 1) \xrightarrow{\text{th}} (\emptyset; \phi; i_f)\)
Class of e-voting protocols

(Honest) Roles:

- **Voter**: \( V(\text{alice}, \text{ballot}) = i : \text{new } \tilde{n}.V' \) such that \( V' \) has no \!, | or new
- **A \in \mathcal{R} authority session**, same format
- **Some role** \( A_c \in \mathcal{R} \) is the bulletin box and \( A_b \in \text{out}(c_b, t) \) “stores in BB”

Tally:

- Made of a public term \( \Psi_b \) (correct form?) and private term Extract (check validity and extract vote)
- ”Tally” = \(!i_f : \text{in}(c, x).\text{let } (_, v) = (\Psi_b[x], \text{Extract}[x]) \text{ in } \text{out}(c, v)\)

Honest Trace: (symbolic) trace th s.t. \( (\mathcal{R} \cup \{V(\text{alice}, \checkmark)\}; \phi_0; 1) \xrightarrow{th} (\emptyset; \phi; i_f) \)

E-Voting Protocol: \( (V; \phi_0; V(\text{alice}, \text{ballot}), \mathcal{R}, (\Psi_b, \text{Extract})) \)
Ballot Secrecy

\[ V(\bigcirc, \checkmark) \mid V(\bigcirc, \times) \mid !\mathcal{R} \mid \text{Tally} \approx? \quad V(\bigcirc, \times) \mid V(\bigcirc, \checkmark) \mid !\mathcal{R} \mid \text{Tally} \]

(Weak) phases are not enough

Take:

\[ V(\bigcirc, \boxdot) = 1 : \text{out}(c, \bigcirc) \]  
\[ 2 : \text{out}(c, \boxdot) \]

In \[ V(\bigcirc, \checkmark) \mid V(\bigcirc, \times) \], \(\bigcirc\) can block \(\bigcirc\) and observes \(\bigcirc\)'s \(\boxdot\): \(\checkmark \neq \times\)

But strong phases suffer from theoretical limitations w.r.t. replications.

Idea:

- Executions with strong phases = executions with weak phases that wait for all processes at each phase jump
Ballot Secrecy

\[ V(\bigcirc, \checkmark) \mid V(\bigcirc, \times) \mid !R \mid \text{Tally} \approx \text{fair} \mid V(\bigcirc, \times) \mid V(\bigcirc, \checkmark) \mid !R \mid \text{Tally} \]

(Weak) phases are not enough

Take: \[ V(\bigcirc, \exists) = 1: \text{out}(c, \bigcirc). \quad 2: \text{out}(c, \exists) \]

In \[ V(\bigcirc, \checkmark) \mid V(\bigcirc, \times) \], \(\exists\) can block \(\bigcirc\) and observes \(\bigcirc\)'s \(\exists\): \(\checkmark \neq \times\)

But strong phases suffer from theoretical limitations w.r.t. replications.

Idea:

- Executions with strong phases = executions with weak phases that wait for all processes at each phase jump

    \[ \cap \]

- Fair executions = executions with weak phases that wait for \(\bigcirc\) and \(\bigcirc\)

Ballot Secrecy: Use weak phases\(\approx_{\text{fair}}\) instead of strong phases\(\approx\)
Leaking Status

\[ V = 1: \begin{array}{|c|c|c|} \hline \text{Out}(\mathbb{H}) & \text{In}(y) & \text{Out}(u) \\ \hline \end{array} \begin{array}{|c|c|} \hline \text{Out}(\mathbb{V}) & \text{Out}(n) \\ \hline \end{array} \]

\[ A = 1: \begin{array}{|c|c|c|} \hline \text{In}(x) & \text{Out}(t) & \text{In}(z) \\ \hline \end{array} \]

But \text{diff-equivalence} is still problematic

Lucca Hirschi & Cas Cremers

Improving Automated Symbolic Analysis of Ballot Secrecy for E-voting Protocols
Leaking Status

\[ V = 1: \begin{array}{c|c|c}
\text{Out}(\text{id}) & \text{In}(y) & \text{Out}(u) \\
\end{array} \quad \begin{array}{c|c}
\text{Out}(\text{v}) & \text{Out}(n) \\
\end{array} \]

\[ A = 1: \begin{array}{c|c|c}
\text{In}(x) & \text{Out}(t) & \text{In}(z) \\
\end{array} \]

But diff-equivalence is still problematic
Leaking Status

\[ V = 1: \]
\[ \text{Out}(\text{id}) \quad \text{In}(y) \quad \text{Out}(u) \]
\[ A = 1: \]
\[ \text{In}(x) \quad \text{Out}(t) \quad \text{In}(z) \]

\[ 2: \quad \text{Out}(\checkmark) \quad \text{Out}(n) \]

But \( \text{diff-equivalence is still problematic} \)

Lucca Hirschi & Cas Cremers
Improving Automated Symbolic Analysis of Ballot Secrecy for E-voting Protocols
Leaking Status

\[ V=1: \quad \text{Out}(\text{id}) \quad \text{In}(y) \quad \text{Out}(u) \]

\[ A=1: \quad \text{In}(x) \quad \text{Out}(t) \quad \text{In}(z) \]

\[ 2: \quad \text{Out}(\text{id}) \quad \text{Out}(n) \]

\text{id-leaking phase} \quad \text{vote-leaking phase}

But \text{diff-equivalence} \text{ is still problematic}
Leaking Status

\[
\begin{align*}
V=1: & \quad \text{Out}(\text{id}) \quad \text{In}(y) \quad \text{Out}(u) \\
A=1: & \quad \text{In}(x) \quad \text{Out}(t) \quad \text{In}(z) \\
\end{align*}
\]

\[\text{id-leaking phase} \quad \text{vote-leaking phase}\]

\[\text{id-leaking phases unlinkable to } v \quad \land \quad \text{vote-leaking phases unlinkable to } \text{id}\]

\[\text{diff}[v_1, v_2] \text{ in } \text{id-leaking phases} \quad \approx \quad \text{diff}[id_1, id_2] \text{ in } \text{vote-leaking phases}\]

\[\Rightarrow \text{ phase leaking status}\]

But \[\text{diff-equivalence is still problematic}\]
Leaking Status

\[ V = 1: \]

\[ A = 1: \]

\[ \text{id-leaking phase} \]

\[ \text{vote-leaking phase} \]

- at most 1 type of leak in a single phase \( \sim \) phase leaking status
  \[
  \begin{align*}
  \text{id-leaking phases unlinkable to } v \\
  \land 
  \text{vote-leaking phases unlinkable to } \text{id}
  \end{align*}
  \approx
  \begin{align*}
  \text{diff}[v_1, v_2] \text{ in id-leaking phases} \\
  \text{diff}[id_1, id_2] \text{ in vote-leaking phases}
  \end{align*}
  \]

- name has at most 1 type of link \( \sim \) name leaking status
  \[
  \begin{align*}
  \text{id-leaking phases/names unlinkable to } v \\
  \land 
  \text{vote-leaking phases/names unlinkable to } \text{id}
  \end{align*}
  \approx
  \begin{align*}
  \text{diff}[n^v_1, n^v_2] \text{ in id-leaking phases} \\
  \text{diff}[n^{id}_1, n^{id}_2] \text{ in vote-leaking phases}
  \end{align*}
  \]

But diff-equivalence is still problematic
Leaking Status

\( V = 1: \)

\[
\begin{array}{ccc}
\text{Out}(\text{id}) & \text{In}(y) & \text{Out}(u) \\
\end{array}
\]

\( A = 1: \)

\[
\begin{array}{ccc}
\text{In}(x) & \text{Out}(t) & \text{In}(z) \\
\end{array}
\]

\( 2: \)

\[
\begin{array}{cc}
\text{Out}(v) & \text{Out}(n) \\
\end{array}
\]

- at most 1 type of leak in a single phase \( \leadsto \) phase leaking status

id-leaking phases unlinkable to \( v \)
\( \land \) vote-leaking phases unlinkable to id

\( \approx \) \( \text{diff}[v_1, v_2] \) in id-leaking phases
\( \text{diff}[id_1, id_2] \) in vote-leaking phases

- name has at most 1 type of link \( \leadsto \) name leaking status

id-leaking phases/names unlinkable to \( v \)
\( \land \) vote-leaking phases/names unlinkable to id

\( \approx \) \( \text{diff}[n^v_1, n^v_2] \) in id-leaking phases
\( \text{diff}[n^{id}_1, n^{id}_2] \) in vote-leaking phases

But diff-equivalence is still problematic
Phase-Process and Dishonest Condition

Idea: if a deviation from the honest execution in phase $i$ has some impact in phase $j > i \leadsto \mathbb{W}$ may link phases $i$ and $j$.

e.g. “weaken”/taint credential in phase 1 and observe it in phase 2

Dishonest Condition (Informal)

For any fair execution $(S; \phi_0; 1) \xrightarrow{t.\text{phase}(j)} (P; \phi; j)$, if a process at phase $j$
annotated $[\mathbb{W}, \mathbb{M}]$ for $\mathbb{W} \in \{\mathbb{W}, \mathbb{M}\}$ and $\mathbb{M} \in \mathcal{V}$ is present in $P$ then it followed th up to phase $j$. 

Lucca Hirschi & Cas Cremers
Improving Automated Symbolic Analysis of Ballot Secrecy for E-voting Protocols
Phase-Process and Dishonest Condition

Idea: if a deviation from the honest execution in phase $i$ has some impact in phase $j > i \sim$ may link phases $i$ and $j$.

e.g. “weaken”/taint credential in phase 1 and observe it in phase 2

Dishonest Condition (Informal)

For any fair execution $(S; \phi_0; 1) \xrightarrow{t \cdot \text{phase}(j)} (P; \phi; j)$, if a process at phase $j$ annotated $[\text{ }, ]$ for $\in \{\text{ }, \text{ }\}$ and $\in V$ is present in $P$ then it followed th up to phase $j$.

- Prevent a class of attacks
- Allow us to focus on less executions (those that meet the condition)

\[ V = 1: \quad \text{Out(id)} | \text{In(y)} | \text{Out(u)} \]

\[ A = 1: \quad \text{In(x)} | \text{Out(t)} | \text{In(z)} \]

\[ 2: \quad \text{Out(V)} | \text{Out(n)} \]
Phase-Process and Dishonest Condition

Idea: if a deviation from the honest execution in phase \( i \) has some impact in phase \( j > i \), then may link phases \( i \) and \( j \).

e.g. “weaken”/taint credential in phase 1 and observe it in phase 2

**Dishonest Condition (Informal)**

For any fair execution \((S; \phi_0; 1) \xrightarrow{t.\text{phase}(j)} (P; \phi; j)\), if a process at phase \( j \) annotated \([\text{\textcolor{red}{\circle{1}}}, \text{\textcolor{red}{\bullet}}]\) for \( \text{\textcolor{red}{\circle{1}}} \in \{\text{\textcolor{red}{\circle{1}}}, \text{\textcolor{red}{\bullet}}\} \) and \( \text{\textcolor{red}{\bullet}} \in V \) is present in \( P \) then it followed th up to phase \( j \).

- Prevent a class of attacks
- Allow us to focus on less executions (those that meet the condition)

\[
\begin{align*}
V = 1: & \quad \text{Out(\text{\textcolor{red}{\bullet}})}, \ \text{In}(y), \ \text{Out}(u) \\
A = 1: & \quad \text{In}(x), \ \text{Out}(t), \ \text{In}(z) \\
& \quad \text{Out(\text{\textcolor{red}{\circle{1}}})}, \ \text{Out}(n)
\end{align*}
\]
Phase-Process and Dishonest Condition

Idea: if a deviation from the honest execution in phase $i$ has some impact in phase $j > i \sim$ may link phases $i$ and $j$.

e.g. “weaken”/taint credential in phase 1 and observe it in phase 2

Dishonest Condition (Informal)

For any fair execution $(S; \phi_0; 1) \xrightarrow{t.\text{phase}(j)} (P; \phi; j)$, if a process at phase $j$ annotated $[\text{\textbullet}, \text{\textclubsuit}]$ for $\text{\textbullet} \in \{\text{\textbullet}, \text{\textclubsuit}\}$ and $\text{\textclubsuit} \in \mathcal{V}$ is present in $P$ then it followed th up to phase $j$.

- Prevent a class of attacks
- Allow us to focus on less executions (those that meet the condition)

\begin{align*}
\mathcal{R}^\text{id}(n^\text{id}_A, n^\text{v}_1) &= \{ 1: \text{Out(\text{\textbullet}1)} \text{ ln(y) out(u) }, \\
&\quad 1: \text{ln(x) out(t) ln(z)} \}
\end{align*}

\begin{align*}
\mathcal{R}^\text{v}(n^\text{id}_A, n^\text{v}_1) &= \{ 2: \text{Out(1) out(n_1) } \}
\end{align*}
Relation Condition

We would like to check the absence of relation for all phase-processes.

(less structural links now 😊)

diff[n^v_3, n^v_5] in id-leaking process-phases

diff[n^{id}_3, n^{id}_5] in vote-leaking process-phases

Formally defined through a bi-process:

\[ B = \{ R^{id}(n^{id}, \text{diff}[n^v_3, n^v_5]), R^v(\text{diff}[n^{id}_3, n^{id}_5], n^v), R^{id}(n^v_3, \text{diff}[n^{id}_5, n^v_3]), R^v(\text{diff}[n^{id}_3, n^{id}_5], n^v_5) \} \]
\[ \cup !R; \phi_0; 1 \]
Relation Condition

We would like to check the absence of relation for all phase-processes. (less structural links now 😊)

diff[\text{\textit{id}}^\text{\textit{id}}, \text{\textit{id}}^\text{\textit{id}}] in \textit{id}-leaking process-phases

diff[\text{\textit{vote}}^\text{\textit{id}}, \text{\textit{vote}}^\text{\textit{id}}] in \textit{vote}-leaking process-phases

Formally defined through a bi-process:

\begin{align*}
B &= \left\{ \mathcal{R}^\text{id}(\text{\textit{id}}^\text{\textit{id}}, \text{\textit{id}}^\text{\textit{id}}, \text{\textit{id}}^\text{\textit{id}}), \mathcal{R}^\text{\textit{vote}}(\text{\textit{id}}^\text{\textit{id}}, \text{\textit{id}}^\text{\textit{id}}, \text{\textit{id}}^\text{\textit{id}}), \\
&\quad \mathcal{R}^\text{id}(\text{\textit{id}}^\text{\textit{id}}, \text{\textit{id}}^\text{\textit{id}}, \text{\textit{vote}}^\text{\textit{id}}), \mathcal{R}^\text{\textit{vote}}(\text{\textit{id}}^\text{\textit{id}}, \text{\textit{id}}^\text{\textit{id}}, \text{\textit{vote}}^\text{\textit{id}}) \right\} \\
& \cup !\mathcal{R}; \phi_0; 1
\end{align*}

Relation Condition (Informal)

The Honest Relations Condition is satisfied if \( B \) is \textit{diff-equivalent} and \( \text{\textit{th}} \) is \textit{phase-oblivious}.

\( \text{\textit{th}} \) is \textit{phase-oblivious} when it does not connect a handle and a recipe of different leaking status
Tally Condition

**Goal:** prevents ballot secrecy attacks that exploit the tally’s outcome.

Ballots are either:

1. **(honest):** stems from an honest execution of 🎅 or 🤴
2. **(dishonest):** does not depend on data that can be linked to an identity
   - the vote Tally would extract is insensible to the swap 🎅 ↔ 🤴
### Tally Condition

**Goal:** prevents ballot secrecy attacks that exploit the tally’s outcome.

Ballots are either:

1. *(honest)*: stems from an honest execution of 🎅 or 🎅
2. *(dishonest)*: does not depend on data that can be linked to an identity
   - the vote Tally would extract is insensible to the swap 🎅 ↔ 🎅

---

### Tally Condition (Informal)

∀ fair execution $B \xrightarrow{t} (\mathcal{P}', (\phi_l, \phi_r))$, for any ballot $w\phi_l$ in the BB, either:

1. there exists a voter $V(办实事, 🎅​), 办实事 \in \{办实事, 🎅​\}$ who had an honest interaction and who has cast $w$
2. or there exists some $v \in V \cup \{⊥\}$ such that $\text{Extract}(w\phi_l) \downarrow v$ and $\text{Extract}(w\phi_r) \downarrow v$. 
Tally Condition

Goal: prevents ballot secrecy attacks that exploit the tally’s outcome.
Ballots are either:

1. **(honest)**: stems from an honest execution of 🎅 or 🎄
2. **(dishonest)**: does not depend on data that can be linked to an identity

~ the vote Tally would extract is insensible to the swap 🎅 ↔ 🎄

Tally Condition (Informal)

∀ fair execution $B^t \rightarrow (P', (\phi_l, \phi_r))$, for any ballot $w\phi_l$ in the BB, either:

1. there exists a voter $V(🎅, ⚪️), ⚪️ \in \{🎅, 🎄\}$ who had an honest interaction and who has cast $w$
2. or there exists some $v \in V \cup \{\perp\}$ such that $\text{Extract}(w\phi_l) \downarrow v$ and $\text{Extract}(w\phi_r) \downarrow v$.

2. Ballot can depend on data from vote-leaking phases but not from id-leaking phases

~ bias leaking information on a ballot unlinkable to 🎅 or 🎄 is ok

~ refines ballot independence
Our Results

Theorem (soundness)

For any $E = (\mathcal{V}; \phi_0; V(\mathcal{H}, \mathcal{E}), \mathcal{R}, (\Psi_b, \text{Extract}))$, if the Dishonest, Relation and Tally conditions hold then $E$ satisfies ballot secrecy.
Our Results

**Theorem (soundness)**

For any $E = (V; \phi_0; V(\text{发声}, \text{投票}), R, (\Psi_b, \text{Extract}))$, if the Dishonest, Relation and Tally conditions hold then $E$ satisfies ballot secrecy.

- We provide an algorithm for computing models checking the conditions and heuristics to find leaking status (checkability) (tool is FW)
- We apply our techniques to several case studies and compare ourselves with the swapping technique (tightness): 

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Ballot Secrecy</th>
<th>Our verif. time</th>
<th>Swapping technique verif. time</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOO</td>
<td>✓</td>
<td>0.04</td>
<td>0.26</td>
</tr>
<tr>
<td>Lee 1</td>
<td>✓</td>
<td>0.04</td>
<td>46</td>
</tr>
<tr>
<td>Lee 2</td>
<td>✓</td>
<td>0.05</td>
<td>†</td>
</tr>
<tr>
<td>Lee 3</td>
<td>✓</td>
<td>0.01</td>
<td>†</td>
</tr>
<tr>
<td>Lee 4</td>
<td>✗</td>
<td>6.64</td>
<td>169.94</td>
</tr>
<tr>
<td>JCJ</td>
<td>✓</td>
<td>18.79</td>
<td>✗</td>
</tr>
<tr>
<td>Belenios</td>
<td>✓</td>
<td>0.02</td>
<td>✗</td>
</tr>
</tbody>
</table>
Conclusion

Reusing core ideas

▸ Adapt for the case of receipt-freeness and coercion-resistance
▸ Reuse methodology for other contexts/privacy properties
▸ Infer generic framework (e.g. separation btw. data and active deviation issues)
▸ Extract guidelines for privacy from our conditions (?)

Future Work

▸ Extend our result with more precise Tally:
▸ Combine with the new BPRIV privacy definition [S&P’15, Euro S&P’19]
▸ Provide a tool with ProVerif/Tamarin as back-end
▸ Reuse methodology for other contexts/privacy properties