Voting: You Can’t Have Privacy without Individual Verifiability

Véronique Cortier, Joseph Lallemand

March 1, 2019
Introduction: e-voting protocols

▶ Using computers to organise elections
    → voting machines in polling stations
    → remote voting on the Internet

▶ More convenient
    → for voters: vote from home, or abroad
    → for authorities: easier to record and tally votes

▶ Many protocols have been proposed:
   Helios, Belenios, Civitas, Prêt-à-Voter,…

▶ But of course:
   need to ensure voting protocols are secure
Voting protocols

What does it mean for a voting protocol to be secure?
Voting protocols

What does it mean for a voting protocol to be secure?
What does it mean for a voting protocol to be secure?
E-voting: security properties

Several properties have been defined:

- **privacy:**
  no one should know who I voted for

- **verifiability:**
  everyone can ensure that the votes are correctly counted

- **receipt-freeness/coercion resistance:**
  even if I want to, I can’t prove who I voted for to someone else

- ...
Vote privacy

▶ What does it mean for the vote to be private?

▶ An attacker is unable to tell *who voted for who*

▶ Indistinguishability property
Verifiability

Divided into three subproperties:

- **individual verifiability:**
  I can check that my vote is in the ballot box

- **universal verifiability:**
  everyone can check that the result corresponds to the ballot box

- **eligibility verifiability:**
  every ballot in the box was cast by a legitimate, registered voter
Privacy vs Verifiability

The two properties seem opposed:

- Privacy: give *no information* about how people voted
- Verifiability: give *enough information* to check each vote is counted
Privacy vs Verifiability

The two properties seem opposed:

- **Privacy**: give *no information* about how people voted
- **Verifiability**: give *enough information* to check each vote is counted

- **Impossibility result**: [Chevallier-Mames, Fouque, Pointcheval, Stern, Traoré, 2010]

  *unconditional privacy and verifiability are incompatible*

  *(i.e. for an attacker with unbounded computing power)*
Privacy vs Verifiability

The two properties seem opposed:

- Privacy: give *no information* about how people voted
- Verifiability: give *enough information* to check each vote is counted

- Impossibility result: [Chevallier-Mames, Fouque, Pointcheval, Stern, Traoré, 2010]
  
  *unconditional privacy and verifiability are incompatible*
  
  *(i.e. for an attacker with unbounded computing power)*

- Regulations choose one over the other
  
  *Ex: in France or Switzerland, privacy is prioritised over verifiability*
Our result

Theorem (informal)

We show that, in fact,

\[ \text{Privacy} \implies \text{Individual Verifiability} \]

- Counter-intuitive, but does not contradict previous impossibility result
  \[ \implies \text{our result is for a polynomial attacker} \]

- How is it possible that some protocols are known to be private and non verifiable?

- What does this tell us about privacy?
Computational model

**Voting scheme:**

\[(\text{Setup}, \text{Vote}, \text{VerifVoter}, \text{Tally}, \text{Valid})\]

- **Setup**\((1^\lambda)\): generate the *election keys* \((pk, sk)\)
- **Vote**\((id, pk, v)\): construct a ballot containing the vote \(v\) for voter \(id\)
- **VerifVoter**\((id, L, BB)\): voter \(id\) checks her vote is counted in \(BB\)
- **Tally**\((BB, sk)\): compute the tally of the ballots on the board \(BB\)
- **Valid**\((id, b, BB, pk)\): checks that a ballot \(b\) cast by \(id\) is valid w.r.t. \(BB\)

counting function \(\rho\): votes \(\rightarrow\) result

with *partial tallying*:\( \forall A, B. \rho(A \uplus B) = \rho(A) \ast \rho(B)\)

**Ex:** multiset, sum, ...
Privacy: game-based definition

Privacy is defined as a cryptographic game

\[ \text{Exp}_{\mathcal{A}}^{\text{priv},\beta}(\lambda) \]
\[(pk, sk) \leftarrow \text{Setup}(1^\lambda)\]
\[\mathcal{A}_1^{\mathcal{O}_{\text{vote}},\mathcal{O}_{\text{cast}}}(pk)\]
\[\text{if } \rho(V_0) = \rho(V_1) \text{ then}\]
\[r \leftarrow \text{Tally}(BB, sk)\]
\[\text{return } \mathcal{A}_2(pk, r)\]

\[\mathcal{O}_{\text{vote}}^{\beta}(id, v_0, v_1)\]
\[b \leftarrow \text{Vote}(id, pk, v_\beta)\]
\[BB \leftarrow BB \parallel b\]
\[V_0 \leftarrow V_0 \parallel v_0\]
\[V_1 \leftarrow V_1 \parallel v_1\]
\[\text{return } b\]

\[\mathcal{O}_{\text{cast}}^{\beta}(id, b)\]
\[\text{if } \text{Valid}(id, b, BB, pk) \text{ then}\]
\[BB \leftarrow BB \parallel b\]

Advantage of the adversary:
\[|P[\text{Exp}_{\mathcal{A}}^{\text{priv},0}(\lambda) = 1] - P[\text{Exp}_{\mathcal{A}}^{\text{priv},1}(\lambda) = 1]|\]
Privacy: game-based definition

Privacy is defined as a cryptographic game

[Benaloh, 1987]

\[
\begin{align*}
\text{Advantage of the adversary: } & \left| \mathbb{P}\left[ \text{Exp}_A^{\text{priv},0}(\lambda) = 1 \right] - \mathbb{P}\left[ \text{Exp}_A^{\text{priv},1}(\lambda) = 1 \right] \right| \\
\text{Attacker has access to vote and cast oracles}
\end{align*}
\]
Privacy: game-based definition

Privacy is defined as a cryptographic game

\[ \text{Exp}_{A}^{\text{priv}, A}(\lambda) \]

\[(pk, sk) \leftarrow \text{Setup}(1^{\lambda}) \]

A_{1}^{\text{O}_{\text{vote}}, \text{O}_{\text{cast}}}(pk)

if \( \rho(V_0) = \rho(V_1) \) then

\[ r \leftarrow \text{Tally}(BB, sk) \]

return \( A_{2}(pk, r) \)

Advantage of the adversary:

\[ |P[\text{Exp}_{A}^{\text{priv}, 1}(\lambda) = 1] - P[\text{Exp}_{A}^{\text{priv}, 1}(\lambda) = 1]| \]

Cast oracle:
cast \( b \) to the ballot box for dishonest \( id \)

\( \text{O}_{\text{vote}}(id, v_0, v_1) \)

\( b \leftarrow \text{Vote}(id, pk, v_{\beta}) \)

\( BB \leftarrow BB \| b \)

\( V_0 \leftarrow V_0 \| v_0 \)

\( V_1 \leftarrow V_1 \| v_1 \)

return \( b \)

\( \text{O}_{\text{cast}}(id, b) \)

if Valid(id, b, BB, pk) then

\( BB \leftarrow BB \| b \)
Privacy: game-based definition

Privacy is defined as a cryptographic game. The vote oracle:
choose two votes \( v_0, v_1 \) for honest voter \( id \)

\[
\text{Exp}_{\mathcal{A}}^{\text{priv}, \beta} (\lambda) \\
(pk, sk) \leftarrow \text{Setup}(1^\lambda) \\
\mathcal{A}_{1}^{\mathcal{O}_{\text{vote}}, \mathcal{O}_{\text{cast}}}(pk) \\
\text{if } \rho(V_0) = \rho(V_1) \text{ then} \\
\quad r \leftarrow \text{Tally}(BB, sk) \\
\quad \text{return } \mathcal{A}_{2}(pk, r)
\]

\[
\mathcal{O}_{\text{vote}}^{\beta}(id, v_0, v_1) \\
b \leftarrow \text{Vote}(id, pk, v_\beta) \\
BB \leftarrow BB \Vert b \\
V_0 \leftarrow V_0 \Vert v_0 \\
V_1 \leftarrow V_1 \Vert v_1 \\
\text{return } b
\]

\[
\mathcal{O}_{\text{cast}}(id, b) \\
v_\beta \text{ goes to the ballot box}
\]

\[
\text{if } \text{Valid}(id, b, BB, pk) \text{ then} \\
BB \leftarrow BB \Vert b \\
v_0, v_1 \text{ are recorded}
\]

Advantage of the adversary:

\[
\left| \mathbb{P}\left[ \text{Exp}_{\mathcal{A}}^{\text{priv}, 0}(\lambda) = 1 \right] - \mathbb{P}\left[ \text{Exp}_{\mathcal{A}}^{\text{priv}, 1}(\lambda) = 1 \right] \right|
\]
Privacy: game-based definition

Privacy is defined as a cryptographic game [Benaloh, 1987]

\[
\exp_{A}^{\text{priv}, \beta}(\lambda) = (pk, sk) \leftarrow \text{Setup}(1^{\lambda}) \\
\mathcal{A}_{1}^{O_{\text{vote}}, O_{\text{cast}}(pk)}
\]

if \( \rho(V_0) = \rho(V_1) \) then
\[
r \leftarrow \text{Tally}(BB, sk) \\
\text{return } \mathcal{A}_{2}(pk, r)
\]

\[
O_{\text{vote}}^{\beta}(pk, V_0, V_1) = b \leftarrow \text{Vote}(id, pk, v_{\beta}) \\
BB \leftarrow BB \parallel b \\
V_0 \leftarrow V_0 \parallel v_{0} \\
V_1 \leftarrow V_1 \parallel v_{1} \\
\text{return } b
\]

if \( \text{Valid}(id, b, BB, pk) \) then
\[
BB \leftarrow BB \parallel b
\]

Advantage of the adversary:
\[
\left| P\left[ \exp_{A}^{\text{priv}, 0}(\lambda) = 1 \right] - P\left[ \exp_{A}^{\text{priv}, 1}(\lambda) = 1 \right] \right|
\]
Privacy: game-based definition

Privacy is defined as a cryptographic game 

[Benaloh, 1987]

\[ \text{Advantage of the adversary:} \quad \left| \mathbb{P} \left[ \text{Exp}_{A}^{\text{priv,0}}(\lambda) = 1 \right] - \mathbb{P} \left[ \text{Exp}_{A}^{\text{priv,1}}(\lambda) = 1 \right] \right| \]
Privacy: game-based definition

Privacy is defined as a cryptographic game

[Benaloh, 1987]

\[
\text{Exp}_{\mathcal{A}}^{\text{priv}, \beta}(\lambda) = \begin{cases} 
(pk, sk) \leftarrow \text{Setup}(1^\lambda) \\
\mathcal{A}_1^{O_{\text{vote}}, O_{\text{cast}}}(pk) \\
\text{if } \rho(V_0) = \rho(V_1) \text{ then} \\
\quad r \leftarrow \text{Tally}(BB, sk) \\
\quad \text{return } \mathcal{A}_2(pk, r) 
\end{cases}
\]

\[
\mathcal{O}_{\text{vote}}(id, v_0, v_1) = \begin{cases} 
b \leftarrow \text{Vote}(id, pk, v_\beta) \\
BB \leftarrow BB || b \\
V_0 \leftarrow V_0 || v_0 \\
V_1 \leftarrow V_1 || v_1 \\
\text{return } b 
\end{cases}
\]

\[
\mathcal{O}_{\text{cast}}(id, b) = \begin{cases} 
\text{if } \text{Valid}(id, b, BB, pk) \text{ then} \\
\quad BB \leftarrow BB || b 
\end{cases}
\]

Advantage of the adversary:

\[
|P\left[\text{Exp}_{\mathcal{A}}^{\text{priv}, 0}(\lambda) = 1\right] - P\left[\text{Exp}_{\mathcal{A}}^{\text{priv}, 1}(\lambda) = 1\right]|
\]
Individual verifiability: game-based definition

\[ \text{Advantage of the adversary: } \mathbb{P} \left[ \text{Exp}^\text{verif} _\mathcal{A} (\lambda) = 1 \right] \]
Individual verifiability: game-based definition

As before: Attacker has vote and cast oracles

\[
\text{Exp}_{\mathcal{A}}^{\text{verif}}(\lambda) = \begin{cases} 
1 & \text{if } \exists V_c. \ r = \rho(Voted \cup V_c) \\
0 & \text{else}
\end{cases}
\]

\[(pk, sk) \leftarrow \text{Setup}(1^\lambda)\]
\[\mathcal{A}^{O_{\text{vote}}, O_{\text{cast}}}(pk)\]
\[r \leftarrow \text{Tally}(BB, sk)\]
\[\text{if } \exists V_c. \ r = \rho(Voted \cup V_c) \text{ then } \]
\[\text{return } 0\]
\[\text{else return } 1\]

Advantage of the adversary: \[P\left[\text{Exp}_{\mathcal{A}}^{\text{verif}}(\lambda) = 1\right]\]

Joseph Lallemand
Voting: Privacy vs Verifiability
March 1, 2019
Individual verifiability: game-based definition

\[\mathcal{O}_{\text{cast}}: \text{cast ballots (dishonest voters)}\]

\[
\begin{align*}
A \leftarrow & \text{Setup}(1^\lambda) \\
A & \leftarrow \mathcal{O}_{\text{vote}}, \mathcal{O}_{\text{cast}}(pk) \\
r & \leftarrow \text{Tally}(BB, sk) \\
\text{if } \exists V_c. \ r = \rho(Voted \cup V_c) \text{ then} \\
& \quad \text{return } 0 \\
\text{else return } 1
\end{align*}
\]

Advantage of the adversary:

\[
P \left[ \text{Exp}_{\mathcal{A}}^\text{verif} (\lambda) = 1 \right]
\]
Individual verifiability: game-based definition

\[ \mathcal{O}_{\text{cast}}: \text{cast ballots (dishonest voters)} \]

\[ \mathcal{O}_{\text{vote}}: \text{choose honest votes} \rightarrow \text{recorded in Voted} \]

\[ \text{Exp}_{A}^{\text{verif}}(\lambda) \]

\[ (pk, sk) \leftarrow \text{Setup}(1^{\lambda}) \]

\[ A^{\mathcal{O}_{\text{vote}}, \mathcal{O}_{\text{cast}}}(pk) \]

\[ r \leftarrow \text{Tally}(BB, sk) \]

\[ \text{if } \exists V_c. \ r = \rho(\text{Voted} \cup V_c) \text{ then} \]

\[ \text{return } 0 \]

\[ \text{else return } 1 \]

Advantage of the adversary:

\[ P \left[ \text{Exp}_{A}^{\text{verif}}(\lambda) = 1 \right] \]
Individual verifiability: game-based definition

\[
\text{Exp}_{\mathcal{A}}^\text{verif}(\lambda)
\]

\[
(pk, sk) \leftarrow \text{Setup}(1^\lambda)
\]
\[
\mathcal{A}^{O_{\text{vote}}, O_{\text{cast}}}(pk)
\]
\[
r \leftarrow \text{Tally}(BB, sk)
\]
\[
\text{if } \exists V_c. r = \rho(\text{Voted} \cup V_c) \text{ then}
\]
\[
\quad \text{return } 0
\]
\[
\text{else return } 1
\]

Advantage of the adversary:
\[
P \left[ \text{Exp}_{\mathcal{A}}^\text{verif}(\lambda) = 1 \right]
\]
Individual verifiability: game-based definition

\[
\text{Exp}^\text{verif}_A(\lambda)
\]

\[
(pk, sk) \leftarrow \text{Setup}(1^\lambda)
\]

\[
A^{O_{vote}, O_{cast}}(pk)
\]

\[
r \leftarrow \text{Tally}(BB, sk)
\]

\[
\text{if } \exists V_c. r = \rho(Voted \cup V_c) \text{ then return 0}
\]

\[
\text{else return 1}
\]

Result contains at least honest votes? if not: \( A \) wins

Advantage of the adversary: \( P\left[\text{Exp}^\text{verif}_A(\lambda) = 1\right] \)
Main result

Theorem (Privacy implies Individual Verifiability (computational))

\[ \exists A. \ P\left[ \text{Exp}^\text{verif}_A(\lambda) = 1 \right] \text{ not negligible} \implies \]

\[ \exists B. \ P\left[ \text{Exp}^\text{priv,0}_B(\lambda) = 1 \right] - P\left[ \text{Exp}^\text{priv,1}_B(\lambda) = 1 \right] \text{ not negligible.} \]

We also prove the same implication in a symbolic model (process algebra), to show its generality:

Theorem (Privacy implies Individual Verifiability (symbolic))

\[ \forall \alpha, a, b. \ P_{\alpha \cup \{a \rightarrow 0, b \rightarrow 1\}} \approx P_{\alpha \cup \{a \rightarrow 1, b \rightarrow 0\}} \implies \]

\[ \forall \alpha. \ \forall (t.\text{out}(ch_r, x), \phi) \in \text{trace}(P_\alpha). \exists V_c. \ \phi(x) = \rho(Voted(t) \cup V_c). \]
Intuition

Assuming there is an attack on individual verifiability, we construct an attack on privacy.

Intuition:
Intuition

Assuming there is an attack on individual verifiability, we construct an attack on privacy.

Intuition:

- assume that the attacker can break verifiability by turning Alice's vote into 1

Result: \{1\}
Intuition

Assuming there is an attack on individual verifiability, we construct an attack on privacy.

Intuition:

▶ assume that the attacker can break verifiability by turning Alice’s vote into 1

▶ consider an attacker against privacy
Intuition

Assuming there is an attack on individual verifiability, we construct an attack on privacy.

Intuition:

- Assume that the attacker can break verifiability by turning Alice’s vote into 1
- Consider an attacker against privacy
- The attacker turns Alice’s vote to 1

Result: \{1, Bob’s vote\} → attacker learns Bob’s vote, and breaks privacy

We generalise this idea to any attack on verifiability.
Intuition

Assuming there is an attack on individual verifiability, we construct an attack on privacy.

Intuition:

- assume that the attacker can break verifiability by turning Alice’s vote into 1
- consider an attacker against privacy
- the attacker turns Alice’s vote to 1
- the result is \{1, Bob’s vote\}
Intuition

Assuming there is an attack on individual verifiability, we construct an attack on privacy.

Intuition:

▶ assume that
the attacker can break verifiability
by turning Alice’s vote into 1

▶ consider an attacker against privacy

▶ the attacker turns Alice’s vote to 1

▶ the result is \{ 1, Bob’s vote \}

⇒ the attacker learns Bob’s vote, and breaks privacy
Intuition

Assuming there is an attack on individual verifiability, we construct an attack on privacy.

Intuition:

- assume that the attacker can break verifiability by turning Alice’s vote into 1
- consider an attacker against privacy
- the attacker turns Alice’s vote to 1
- the result is \{1, Bob’s vote\}

\[ \Rightarrow \] the attacker learns Bob’s vote, and breaks privacy

We generalise this idea to any attack on verifiability.
Proof sketch (assuming a blank vote)

Assuming $A$ breaks verifiability we build $B$ that breaks privacy.

$A$

$B$

$\beta = 0 \quad \beta = 1$
Proof sketch (assuming a blank vote)
Assuming $A$ breaks verifiability we build $B$ that breaks privacy.

Say $A$ uses voters $id_1, \ldots, id_n$.
$B$ will add $n$ fresh voters: $id_1, \ldots, id_n, id'_1, \ldots, id'_n$. 
Proof sketch (assuming a blank vote)

Assuming $A$ breaks verifiability we build $B$ that breaks privacy.

Say $A$ uses voters $id_1, \ldots, id_n$.
$B$ will add $n$ fresh voters: $id_1, \ldots, id_n, id'_1, \ldots, id'_n$. 
Proof sketch (assuming a blank vote)

Assuming $A$ breaks verifiability we build $B$ that breaks privacy.

- Say $A$ uses voters $id_1, \ldots, id_n$.
  - $B$ will add $n$ fresh voters: $id_1, \ldots, id_n, id'_1, \ldots, id'_n$.
- $B$ simulates $A$.
  - Whenever $A$ makes $id_i$ vote for $v_i$,
    - $B$ makes $id_i$ vote $v_i$ on the left, and $blank$ on the right.
Proof sketch (assuming a blank vote)

Assuming \( A \) breaks verifiability we build \( B \) that breaks privacy.

Say \( A \) uses voters \( id_1, \ldots, id_n \).
\( B \) will add \( n \) fresh voters: \( id_1, \ldots, id_n, id'_1, \ldots, id'_n \).

\( B \) simulates \( A \).
Whenever \( A \) makes \( id_i \) vote for \( v_i \),
\( B \) makes \( id_i \) vote \( v_i \) on the left, and \textit{blank} on the right.
Proof sketch (assuming a blank vote)

Assuming \( A \) breaks verifiability we build \( B \) that breaks privacy.

- Say \( A \) uses voters \( id_1, \ldots, id_n \).
  \( B \) will add \( n \) fresh voters: \( id_1, \ldots, id_n, id'_1, \ldots, id'_n \).
- \( B \) simulates \( A \).
  Whenever \( A \) makes \( id_i \) vote for \( v_i \),
  \( B \) makes \( id_i \) vote \( v_i \) on the left, and blank on the right.
Proof sketch (assuming a blank vote)

Assuming $A$ breaks verifiability we build $B$ that breaks privacy.

At this point, the tally would be

- on the left: some $r$ that does not contain all the $v_i$
- on the right: some $r'$. 
Proof sketch (assuming a blank vote)

Assuming $A$ breaks verifiability we build $B$ that breaks privacy.

At this point, the tally would be
- on the left: some $r$ that does not contain all the $v_i$
- on the right: some $r'$. 
Proof sketch (assuming a blank vote)
Assuming $A$ breaks verifiability we build $B$ that breaks privacy.

At this point, the tally would be
- on the left: some $r$ that does not contain all the $v_i$
- on the right: some $r'$.

$B$ then makes each $id_i$ vote **blank** on the left, and $v_i$ on the right.
Proof sketch (assuming a blank vote)

Assuming $A$ breaks verifiability we build $B$ that breaks privacy.

- The sets of honest votes are the same on both sides: $B$ gets the result.
- The result is:
  - on the left: $r \ast blank^n = r$
  - on the right: $r' \ast v_1 \ast \ldots \ast v_n$
Proof sketch (assuming a blank vote)

Assuming \( A \) breaks verifiability we build \( B \) that breaks privacy.

The sets of honest votes are the same on both sides: \( B \) gets the result.

The result is:

- on the left: \( \star \text{blank}^n = r \)
- on the right: \( r' \star v_1 \star \ldots \star v_n \)
Proof sketch (assuming a blank vote)

Assuming $A$ breaks verifiability we build $B$ that breaks privacy.

$B$ checks if the result contains all the $v_i$:
yes on the right, no on the left.
What do we learn from this result?

- Designing a private voting system without caring for verifiability is hopeless:
  
  you need \textit{at least} individual verifiability
What do we learn from this result?

- Designing a private voting system without caring for verifiability is hopeless:
  
  you need *at least* individual verifiability

- But some protocols are proved private while non verifiable?  
  *Ex:* Helios *without modelling the verification steps*
What do we learn from this result?

- Designing a private voting system without caring for verifiability is hopeless:
  
  you need *at least* individual verifiability

- But some protocols are proved private while non verifiable? 
  Ex: Helios *without modelling the verification steps*

  → Our result:

  Privacy ⇒ Individual verifiability *with the same trust assumptions*
What do we learn from this result?

- Designing a private voting system without caring for verifiability is hopeless:
  
  you need at least individual verifiability

- But some protocols are proved private while non verifiable? 
  
  **Ex:** Helios *without modelling the verification steps*

  → Our result:

  Privacy ⇒ Individual verifiability *with the same trust assumptions*

  → What is usually studied:

  Privacy vs *honest ballot box* but Verifiability vs *dishonest ballot box*
What do we learn from this result?

- Designing a private voting system without caring for verifiability is hopeless:
  
  you need at least individual verifiability

- But some protocols are proved private while non verifiable?  
  
  Ex: Helios without modelling the verification steps

  → Our result:
  
  Privacy ⇒ Individual verifiability with the same trust assumptions

  → What is usually studied:

  Privacy vs honest ballot box but Verifiability vs dishonest ballot box

  But protocols aim for privacy against a dishonest ballot box!
The problem with privacy

Problem with existing game-based definitions:
the ballot box is assumed honest → considerably weakens privacy!
The problem with privacy

Problem with existing game-based definitions: the ballot box is assumed honest → considerably weakens privacy!

Because privacy against a dishonest ballot box is hard: adapting naïvely the definition does not work

A dishonest ballot box can drop every ballot except Alice’s → The result is just Alice’s vote!
The problem with privacy

- Problem with existing game-based definitions: the ballot box is assumed honest → considerably weakens privacy!

- Because privacy against a dishonest ballot box is hard: adapting naïvely the definition does not work

- A dishonest ballot box can drop every ballot except Alice’s → The result is just Alice’s vote!

- We need a new definition of privacy, against a dishonest ballot box
Our proposition: privacy with careful voters

- Privacy is linked with verifiability
  \[ \implies \text{let’s introduce the verification steps of the protocol in privacy!} \]
Our proposition: privacy with careful voters

- Privacy is linked with verifiability
  \[ \implies \text{let’s introduce the verification steps of the protocol in privacy!} \]

- The attacker can’t distinguish who voted for who,
  provided all voters perform the verifications:

\[
\begin{align*}
\text{Exp}_{A}^\text{priv−careful, }\beta (\lambda) \\
(pk, sk) &\leftarrow \text{Setup}(1^\lambda) \\
BB &\leftarrow A^O_{1\text{vote}}(pk) \\
A^O_{\text{happy}} (pk) &
\begin{cases}
\text{if } \forall id \in V_0, V_1. id \in H \land \rho(V_0) = \rho(V_1) \text{ then} & \quad r \leftarrow \text{Tally}(BB, sk) \\
\text{else} & \quad r \leftarrow \bot
\end{cases}
\end{align*}
\]

\[
\begin{align*}
O_{\text{vote}}(id, v_0, v_1) \\
b &\leftarrow \text{Vote}(id, pk, v_\beta) \\
V_i &\leftarrow V_i || v_i \text{ for } i \in \{0, 1\} \\
L_{id} &\leftarrow L_{id} || (b, v_\beta) \\
\text{return } b
\end{align*}
\]

\[
\begin{align*}
O_{\text{happy}}^\text{BB}(id) \\
\text{if } \text{VerifVoter}(id, L_{id}, BB) \text{ then} & \quad H \leftarrow H || id
\end{align*}
\]
Our proposition: privacy with careful voters

- Privacy is linked with verifiability
  
  \[ \text{let's introduce the verification steps of the protocol in privacy!} \]

- The attacker can’t distinguish who voted for who, provided all voters perform the verifications:

\[
\begin{align*}
\text{Exp}_{\mathcal{A}}^{\text{priv-careful}, \beta} (\lambda) & \quad (pk, sk) \leftarrow \text{Setup}(1^\lambda) \\
\mathcal{BB} & \leftarrow \mathcal{A}_1^{O_{\text{vote}}}(pk) \\
\mathcal{A}_2^{O_{\text{happy}}}(pk) & \\
\text{if } \forall id \in V_0, V_1. id \in H \land \rho(V_0) = \rho(V_1) \text{ then} \\
& \quad r \leftarrow \text{Tally} (\mathcal{BB}, sk) \\
\text{else } r & \leftarrow \perp \\
\text{return } \mathcal{A}_3(pk, r)
\end{align*}
\]

\[
\begin{align*}
\text{O}_{\text{vote}}(id, v_0, v_1) & \quad b \leftarrow \text{Vote}(id, pk, v_\beta) \\
\mathcal{V}_i & \leftarrow V_i \| v_i \text{ for } i \in \{0, 1\} \\
\mathcal{L}_{id} & \leftarrow \mathcal{L}_{id} \| (b, v_\beta) \\
\text{return } b \\
\text{O}_{\text{happy}}^{BB}(id) & \quad \text{if VerifVoter}(id, \mathcal{L}_{id}, \mathcal{BB}) \text{ then} \\
& \quad H \leftarrow H \| id
\end{align*}
\]
Our proposition: privacy with careful voters

- Privacy is linked with verifiability

\[ \implies \text{let's introduce the verification steps of the protocol in privacy!} \]

- The attacker can't distinguish who voted for who, provided all voters perform the verifications:

\[
\begin{align*}
\text{Exp}^{\text{priv careful, } \beta}_{\mathcal{A}}(\lambda) \\
(pk, sk) \leftarrow \text{Setup}(1^\lambda) \\
\mathcal{B} \leftarrow \mathcal{A}_1^{\mathcal{O}_{\text{vote}}}(pk) \\
\mathcal{A}_2^{\mathcal{O}_{\text{BB happy}}}(pk) \\
\text{if } \forall id \in V_0, V_1. \ id \in H \land \rho(V_0) = \rho(V_1) \text{ then} \\
\quad r \leftarrow \text{Tally}(\mathcal{B}, sk) \\
\text{else } r \leftarrow \bot \\
\text{return } \mathcal{A}_3(pk, r)
\end{align*}
\]

\[O_{\text{vote}}(id, v_0, v_1)\]

\[b \leftarrow \text{Vote}(id, pk, v_\beta)\]

\[V_i \leftarrow V_i \parallel v_i \text{ for } i \in \{0, 1\}\]

\[L_{id} \leftarrow L_{id} \parallel (b, v_\beta)\]

\[\text{return } b\]

\[\mathcal{O}_{\text{happy}}(id)\]

\[\text{if VerifVoter}(id, L_{id}, \mathcal{B}) \text{ then}\]

\[H \leftarrow H \parallel id\]
Our proposition: privacy with careful voters

- Privacy is linked with verifiability
  
  ⇒ let’s introduce the verification steps of the protocol in privacy!

- The attacker cannot distinguish who voted for who, provided all voters perform the verifications:

\[
\begin{align*}
\text{Exp}_{\mathcal{A}}^{\text{priv-careful,} \beta} (\lambda) \\
(pk, sk) \leftarrow \text{Setup}(1^\lambda) \\
\mathcal{B} \leftarrow \mathcal{A}_1^{\mathcal{O}_{\text{vote}}}(pk) \\
\mathcal{A}_2^{\mathcal{O}_{\text{happy}}}(pk) \\
\text{if } \forall id \in V_0, V_1. \ id \in H \land \rho(V_0) = \rho(V_1) \text{ then} \\
\quad r \leftarrow \text{Tally}(\mathcal{B}, sk) \\
\text{else } r \leftarrow \bot \\
\text{return } \mathcal{A}_3(pk, r)
\end{align*}
\]
Our proposition: privacy with careful voters

- Privacy is linked with verifiability
  $\Rightarrow$ let’s introduce the verification steps of the protocol in privacy!

- The attacker can’t distinguish who voted for who, provided all voters perform the verifications:

\[
\text{Exp}_{\mathcal{A}}^{\text{privacy-careful},\beta}(\lambda)
\]

\[
(pk, sk) \leftarrow \text{Setup}(1^\lambda)
\]

\[
\mathcal{B} \leftarrow \mathcal{A}_1^{\mathcal{O}_\text{vote}}(pk)
\]

\[
\mathcal{A}_2^{\mathcal{O}_\text{happy}}(pk)
\]

\[
\text{if } \forall id \in V_0, V_1. id \in H \land \rho(V_0) = \rho(V_1) \text{ then }
\]

\[
r \leftarrow \text{Tally}(BB, sk)
\]

\[
\text{return } \mathcal{A}_3(pk, r)
\]

\[
b \leftarrow \text{Vote}(id, pk, v_\beta)
\]

\[
V_i \leftarrow V_i \| v_i \text{ for } i \in \{0, 1\}
\]

\[
L_id \leftarrow L_id \| (b, v_\beta)
\]

\[
\text{return } b
\]

\[
\mathcal{O}_\text{happy}^{\mathcal{BB}}(id)
\]

\[
\text{if } \text{VerifVoter}(id, L_id, BB) \text{ then }
\]

\[
H \leftarrow H \| id
\]
Our proposition: privacy with careful voters

- Our result still holds for our new definition:

### Theorem

Privacy against a dishonest ballot box with careful voters \( \implies \)

Individual Verifiability against a dishonest ballot box

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Honest box</th>
</tr>
</thead>
<tbody>
<tr>
<td>naïve</td>
<td>[\text{attack P. Roenne}]</td>
</tr>
<tr>
<td>Helios</td>
<td></td>
</tr>
<tr>
<td>Belenios</td>
<td></td>
</tr>
<tr>
<td>Civitas (no revote)</td>
<td></td>
</tr>
<tr>
<td>Neuchâtel (no revote)</td>
<td></td>
</tr>
</tbody>
</table>
Our proposition: privacy with careful voters

- Our result still holds for our new definition:

**Theorem**

Privacy against a dishonest ballot box with careful voters \( \implies \)

Individual Verifiability against a dishonest ballot box

- We apply it to a few existing protocols, to show its relevance

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Honest box</th>
<th>Dishonest box naïve</th>
<th>Careful voters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Helios</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Belenios</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Civitas (no revote)</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Neuchâtel (no revote)</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>

✓: the protocol is private, ×: attack on privacy
Work in progress: towards more precise definitions

- Privacy with careful voters is a first step, but not enough: only says something when everyone verifies
  = "among people who check, the attacker does not know who voted for who"

- Problem: not easy to have an indistinguishability game for voters who do not check
  = as soon as someone does not check, there is a loss of privacy

- Seems more doable with another way of writing properties
Simulation-based definition

- Idea: describe an ideal system, where the attacker "obviously" has no power
- Prove (reduction) that the ideal attacker can simulate everything the real one can do.
Ideal functionality for voting

Case of a honest ballot box:

Ideal functionality $F_{voting}(\rho)$ interacts with environment $\mathcal{E}$ and simulator $S$. $F_{voting}(\rho)$ accepts two kinds of messages:

- on input $vote(id, v)$ from $\mathcal{E}$ or $S$: store $(id, v)$ in a list $L$, and send $ack(id)$ to $S$.
- on input $tally$ from $S$, return $\rho(L)$ to $\mathcal{E}$ and $S$, then halt.

Clearly, $S$ learns no information on the honest votes.

→ Problem: with a dishonest ballot box, this cannot be realised
→ Need to distinguish between voters who check and others
Conclusion

▶ A counter-intuitive result:
   Privacy \implies Individual Verifiability

▶ Proved in computational and symbolic models

▶ Better understanding of privacy: some verifiability is required!

Highlights limitations of game-based current definitions: only honest ballot boxes [Bernhard, Smyth, 2014]

▶ A new definition of privacy against a dishonest ballot box → modelling verification steps

▶ Limitation: assumes everyone checks their vote → Future work: more plausible scenario where only some voters check
Conclusion

- A counter-intuitive result:
  \[ \text{Privacy} \iff \text{Individual Verifiability} \]

- Proved in computational and symbolic models

- Better understanding of privacy: some verifiability is required!

- Highlights limitations of game-based current definitions: only honest ballot boxes [Bernhard, Smyth, 2014]
Conclusion

- A counter-intuitive result: Privacy $\Rightarrow$ Individual Verifiability
- Proved in computational and symbolic models
- Better understanding of privacy: some verifiability is required!
- Highlights limitations of game-based current definitions: only honest ballot boxes [Bernhard, Smyth, 2014]
- A new definition of privacy against a dishonest ballot box $\Rightarrow$ modelling verification steps
- Limitation: assumes everyone checks their vote $\Rightarrow$ Future work: more plausible scenario where only some voters check