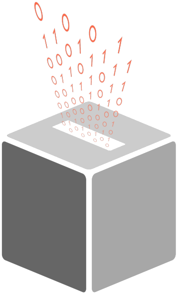
Voting: You Can't Have Privacy without Individual Verifiability

Véronique Cortier, Joseph Lallemand

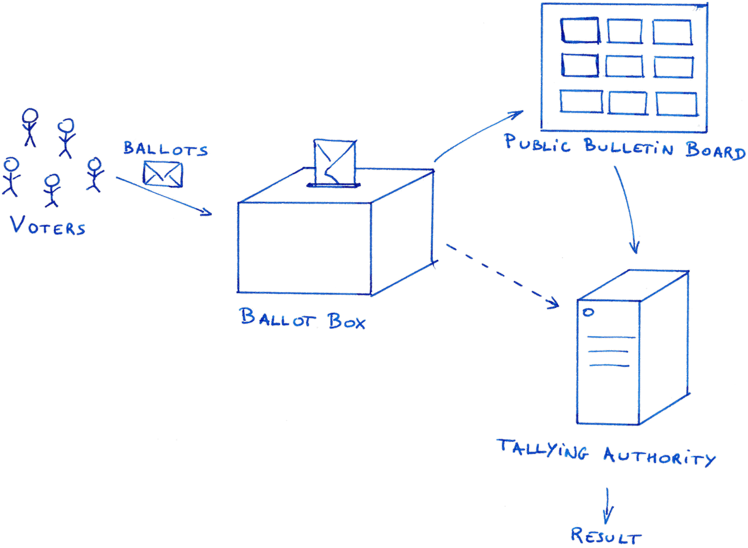
March 1, 2019

Introduction: e-voting protocols

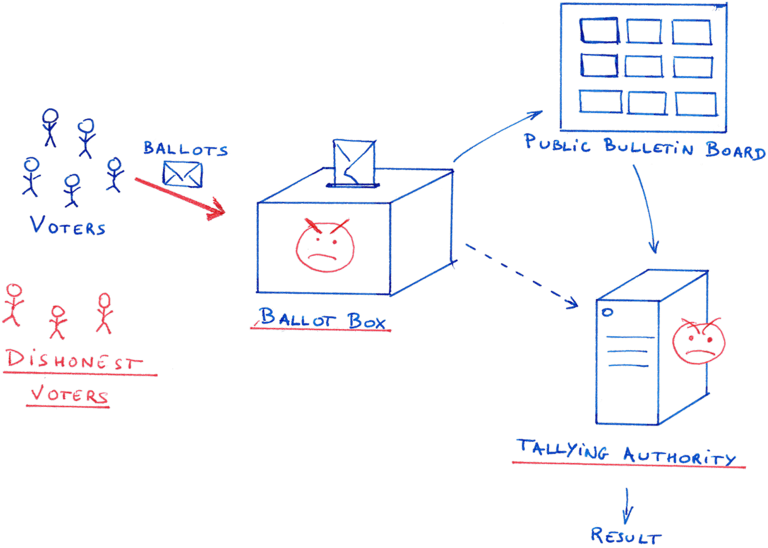
- Using computers to organise elections
  $\longrightarrow$ voting machines in polling stations
  $\longrightarrow$ remote voting on the Internet
- More convenient
  $\longrightarrow$ for voters: vote from home, or abroad
  $\longrightarrow$ for authorities: easier to record and tally votes
- Many protocols have been proposed:
  Helios, Belenios, Civitas, Prêt-à-Voter,...
- But of course:
  need to ensure voting protocols are secure



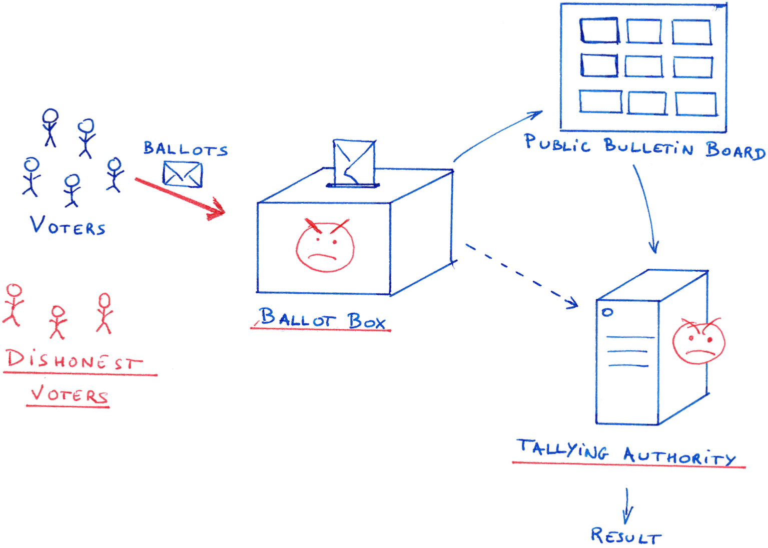
Voting protocols



Voting protocols



Voting protocols



$\Longrightarrow$ What does it mean for a voting protocol to be secure?

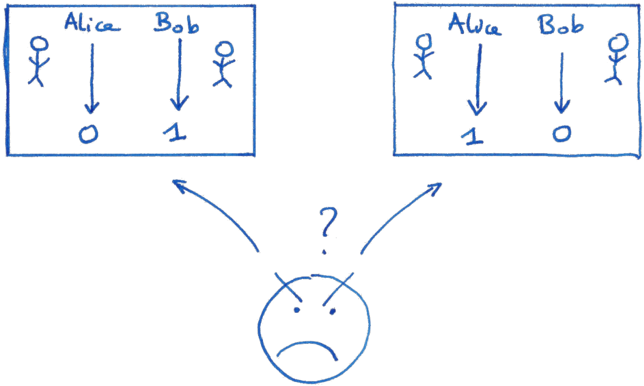
E-voting: security properties

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,

$$\textit{Privacy} \Longrightarrow \textit{Individual Verifiability}$$

- Counter-intuitive, but does not contradict previous impossibility result $\longrightarrow$ 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:

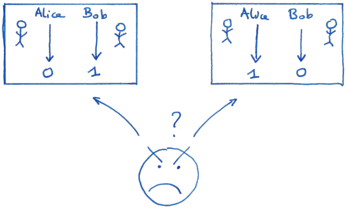
$$(\mathsf{Setup}, \mathsf{Vote}, \mathsf{VerifVoter}, \mathsf{Tally}, \mathsf{Valid})$$

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

counting function ρ: votes $\rightarrow$ result
with *partial tallying*: $\forall A, B.\ \rho(A \uplus B) = \rho(A) * \rho(B)$
Ex: multiset, sum, ...

Privacy: game-based definition

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



$\underline{\mathrm{Exp}_{\mathcal{A}}^{\mathrm{priv},\beta}(\lambda)}$

$(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathrm{Setup}(1^\lambda)$
$\mathcal{A}_1^{\mathcal{O}_{\mathrm{vote}}^{\beta}, \mathcal{O}_{\mathrm{cast}}}(\mathrm{pk})$
if $\rho(\mathrm{V}_0) = \rho(\mathrm{V}_1)$ **then**
  $r \leftarrow \mathrm{Tally}(\mathrm{BB}, \mathrm{sk})$
  return $\mathcal{A}_2(\mathrm{pk}, r)$

$\underline{\mathcal{O}_{\mathrm{vote}}^{\beta}(id, v_0, v_1)}$

$b \leftarrow \mathrm{Vote}(id, \mathrm{pk}, v_\beta)$
$\mathrm{BB} \leftarrow \mathrm{BB} \| b$
$\mathrm{V}_0 \leftarrow \mathrm{V}_0 \| v_0$
$\mathrm{V}_1 \leftarrow \mathrm{V}_1 \| v_1$
return b

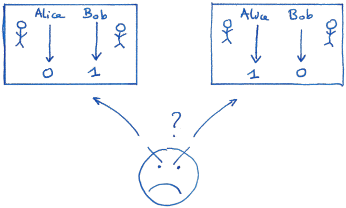
$\underline{\mathcal{O}_{\mathrm{cast}}(id, b)}$

if $\mathrm{Valid}(id, b, \mathrm{BB}, \mathrm{pk})$ **then**
  $\mathrm{BB} \leftarrow \mathrm{BB} \| b$

Advantage of the adversary: $\left| \mathrm{P}\left[\mathrm{Exp}_{\mathcal{A}}^{\mathrm{priv},0}(\lambda) = 1\right] - \mathrm{P}\left[\mathrm{Exp}_{\mathcal{A}}^{\mathrm{priv},1}(\lambda) = 1\right] \right|$

Privacy: game-based definition

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



$\underline{\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv},\beta}(\lambda)}$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$

$\mathcal{A}_1^{\mathcal{O}_{\mathsf{vote}}^{\beta}, \mathcal{O}_{\mathsf{cast}}}(\mathsf{pk})$

if $\rho(\mathsf{V}_0) = \rho(\mathsf{V}_1)$ **then**

  $r \leftarrow \mathsf{Tally}(\mathsf{BB}, \mathsf{sk})$

  return $\mathcal{A}_2(\mathsf{pk}, r)$

$\underline{\mathcal{O}_{\mathsf{vote}}^{\beta}(id, v_0, v_1)}$

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

$\mathsf{BB} \leftarrow \mathsf{BB} \| b$

$\mathsf{V}_0 \leftarrow \mathsf{V}_0 \| v_0$

$\mathsf{V}_1 \leftarrow \mathsf{V}_1 \| v_1$

return b

$\underline{\mathcal{O}_{\mathsf{cast}}(id, b)}$

if $\mathsf{Valid}(id, b, \mathsf{BB}, \mathsf{pk})$ **then**

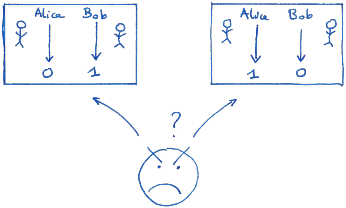
  $\mathsf{BB} \leftarrow \mathsf{BB} \| b$

Attacker has access to vote and cast oracles

Advantage of the adversary: $\left| \mathsf{P}\left[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv},0}(\lambda) = 1\right] - \mathsf{P}\left[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv},1}(\lambda) = 1\right] \right|$

Privacy: game-based definition

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



$\underline{\mathrm{Exp}_{\mathcal{A}}^{\mathrm{priv},\beta}(\lambda)}$

$(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathrm{Setup}(1^{\lambda})$
$\mathcal{A}_1^{\mathcal{O}_{\mathrm{vote}}^{\beta}, \mathcal{O}_{\mathrm{cast}}}(\mathrm{pk})$
if $\rho(\mathrm{V}_0) = \rho(\mathrm{V}_1)$ **then**
  $r \leftarrow \mathrm{Tally}(\mathrm{BB}, \mathrm{sk})$
  return $\mathcal{A}_2(\mathrm{pk}, r)$

$\underline{\mathcal{O}_{\mathrm{vote}}^{\beta}(id, v_0, v_1)}$

$b \leftarrow \mathrm{Vote}(id, \mathrm{pk}, v_{\beta})$
$\mathrm{BB} \leftarrow \mathrm{BB} \| b$
$\mathrm{V}_0 \leftarrow \mathrm{V}_0 \| v_0$
$\mathrm{V}_1 \leftarrow \mathrm{V}_1 \| v_1$
return b

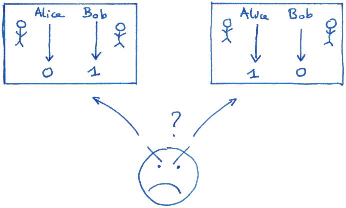
$\underline{\mathcal{O}_{\mathrm{cast}}(id, b)}$

if $\mathrm{Valid}(id, b, \mathrm{BB}, \mathrm{pk})$ **then**
  $\mathrm{BB} \leftarrow \mathrm{BB} \| b$

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

Advantage of the adversary: $\left| \mathrm{P}\left[\mathrm{Exp}_{\mathcal{A}}^{\mathrm{priv},0}(\lambda) = 1\right] - \mathrm{P}\left[\mathrm{Exp}_{\mathcal{A}}^{\mathrm{priv},1}(\lambda) = 1\right] \right|$

Privacy: game-based definition



Privacy is defined as a cryptographic game

Vote oracle:
choose two votes v_0, v_1
for honest voter *id*

$\underline{\mathrm{Exp}_{\mathcal{A}}^{\mathrm{priv},\beta}(\lambda)}$

$(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathrm{Setup}(1^\lambda)$

$\mathcal{A}_1^{\mathcal{O}_{\mathrm{vote}}^{\beta}, \mathcal{O}_{\mathrm{cast}}}(\mathrm{pk})$

if $\rho(\mathrm{V}_0) = \rho(\mathrm{V}_1)$ **then**

$\quad r \leftarrow \mathrm{Tally}(\mathrm{BB}, \mathrm{sk})$

$\quad$ **return** $\mathcal{A}_2(\mathrm{pk}, r)$

$\underline{\mathcal{O}_{\mathrm{vote}}^{\beta}(id, v_0, v_1)}$

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

$\mathrm{BB} \leftarrow \mathrm{BB} \| b$

$\mathrm{V}_0 \leftarrow \mathrm{V}_0 \| v_0$

$\mathrm{V}_1 \leftarrow \mathrm{V}_1 \| v_1$

return b

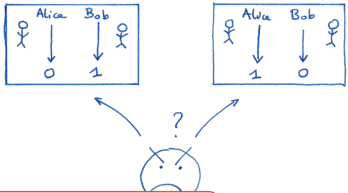
$\underline{\mathcal{O}_{\mathrm{cast}}(id, b)}$

v_β goes to the ballot box

v_0, v_1 are recorded

Advantage of the adversary: $\left| \mathrm{P}\left[\mathrm{Exp}_{\mathcal{A}}^{\mathrm{priv},0}(\lambda) = 1\right] - \mathrm{P}\left[\mathrm{Exp}_{\mathcal{A}}^{\mathrm{priv},1}(\lambda) = 1\right] \right|$

Privacy: game-based definition



Privacy is defined as a cryptographic game

[Benaloh, 1987]

Attacker sees the tally, guesses β

$\underline{\mathsf{Exp}^{\mathsf{priv},\beta}_{\mathcal{A}}(\lambda)}$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$

$\mathcal{A}_1^{\mathcal{O}^\beta_{\mathsf{vote}}, \mathcal{O}_{\mathsf{cast}}}(\mathsf{pk})$

if $\rho(\mathsf{V}_0) = \rho(\mathsf{V}_1)$ **then**

$\quad r \leftarrow \mathsf{Tally}(\mathsf{BB}, \mathsf{sk})$

$\quad$ **return** $\mathcal{A}_2(\mathsf{pk}, r)$

$\underline{\mathcal{O}^\beta_{\mathsf{vote}}(id, v_0, v_1)}$

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

$\mathsf{BB} \leftarrow \mathsf{BB} \| b$

$\mathsf{V}_0 \leftarrow \mathsf{V}_0 \| v_0$

$\mathsf{V}_1 \leftarrow \mathsf{V}_1 \| v_1$

return b

$\underline{\mathcal{O}_{\mathsf{cast}}(id, b)}$

if $\mathsf{Valid}(id, b, \mathsf{BB}, \mathsf{pk})$ **then**

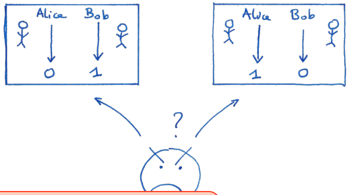
$\quad \mathsf{BB} \leftarrow \mathsf{BB} \| b$

Advantage of the adversary: $\left| \mathsf{P}\left[\mathsf{Exp}^{\mathsf{priv},0}_{\mathcal{A}}(\lambda) = 1\right] - \mathsf{P}\left[\mathsf{Exp}^{\mathsf{priv},1}_{\mathcal{A}}(\lambda) = 1\right] \right|$

Privacy: game-based definition

Privacy is defined as a cryptographic game

[Benaloh, 1987]



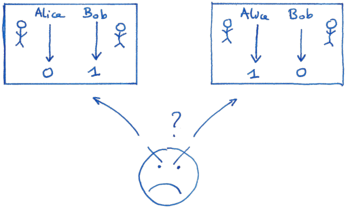
Attacker sees the tally, guesses β

$\mathrm{Exp}^{\mathsf{priv},\beta}_{\mathcal{A}}(\lambda)$	$\mathcal{O}^{\beta}_{\mathsf{vote}}(id, v_0, v_1)$	$\mathcal{O}_{\mathsf{cast}}(id, b)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$	$b \leftarrow \mathsf{Vote}(id, \mathsf{pk}, v_\beta)$	**if** $\mathsf{Valid}(id, b, \mathsf{BB}, \mathsf{pk})$ **then**
$\mathcal{A}_1^{\mathcal{O}^{\beta}_{\mathsf{vote}}, \mathcal{O}_{\mathsf{cast}}}(\mathsf{pk})$	$\mathsf{BB} \leftarrow \mathsf{BB} \| b$	$\quad \mathsf{BB} \leftarrow \mathsf{BB} \| b$
if $\rho(\mathsf{V}_0) = \rho(\mathsf{V}_1)$ **then**	$\mathsf{V}_0 \leftarrow \mathsf{V}_0 \| v_0$	
$\quad r \leftarrow \mathsf{Tally}(\mathsf{BB}, \mathsf{sk})$	$\mathsf{V}_1 \leftarrow \mathsf{V}_1 \| v_1$	
$\quad$ **return** $\mathcal{A}_2(\mathsf{pk}, r)$	**return** b	

Provided the distributions of votes for $\beta = 0, 1$ give the same result

Advantage of the adversary: $\left| \mathrm{P}\left[\mathrm{Exp}^{\mathsf{priv},0}_{\mathcal{A}}(\lambda) = 1\right] - \mathrm{P}\left[\mathrm{Exp}^{\mathsf{priv},1}_{\mathcal{A}}(\lambda) = 1\right] \right|$

Privacy: game-based definition



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

$\underline{\mathrm{Exp}_{\mathcal{A}}^{\mathrm{priv},\beta}(\lambda)}$

$(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathrm{Setup}(1^\lambda)$

$\mathcal{A}_1^{\mathcal{O}_{\mathrm{vote}}^{\beta}, \mathcal{O}_{\mathrm{cast}}}(\mathrm{pk})$

if $\rho(\mathrm{V}_0) = \rho(\mathrm{V}_1)$ **then**

  $r \leftarrow \mathrm{Tally}(\mathrm{BB}, \mathrm{sk})$

  return $\mathcal{A}_2(\mathrm{pk}, r)$

$\underline{\mathcal{O}_{\mathrm{vote}}^{\beta}(id, v_0, v_1)}$

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

$\mathrm{BB} \leftarrow \mathrm{BB} \| b$

$\mathrm{V}_0 \leftarrow \mathrm{V}_0 \| v_0$

$\mathrm{V}_1 \leftarrow \mathrm{V}_1 \| v_1$

return b

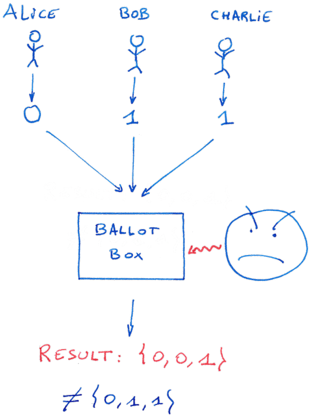
$\underline{\mathcal{O}_{\mathrm{cast}}(id, b)}$

if $\mathrm{Valid}(id, b, \mathrm{BB}, \mathrm{pk})$ **then**

  $\mathrm{BB} \leftarrow \mathrm{BB} \| b$

Advantage of the adversary: $\left| \mathrm{P}\left[\mathrm{Exp}_{\mathcal{A}}^{\mathrm{priv},0}(\lambda) = 1\right] - \mathrm{P}\left[\mathrm{Exp}_{\mathcal{A}}^{\mathrm{priv},1}(\lambda) = 1\right] \right|$

Individual verifiability: game-based definition



$\underline{\mathsf{Exp}^{\mathsf{verif}}_{\mathcal{A}}(\lambda)}$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$

$\mathcal{A}^{\mathcal{O}_{\mathsf{vote}}, \mathcal{O}_{\mathsf{cast}}}(\mathsf{pk})$

$r \leftarrow \mathsf{Tally}(\mathsf{BB}, \mathsf{sk})$

if $\exists V_c.\ r = \rho(\mathsf{Voted} \cup V_c)$ **then**

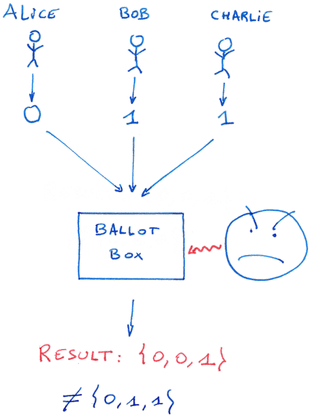
  return 0

else return 1

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

Individual verifiability: game-b

As before:
Attacker has vote and cast oracles



$\underline{\mathsf{Exp}^{\mathsf{verif}}_{\mathcal{A}}(\lambda)}$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$

$\mathcal{A}^{\mathcal{O}_{\mathsf{vote}}, \mathcal{O}_{\mathsf{cast}}}(\mathsf{pk})$

$r \leftarrow \mathsf{Tally}(\mathsf{BB}, \mathsf{sk})$

if $\exists V_c.\ r = \rho(\mathsf{Voted} \cup V_c)$ **then**

  return 0

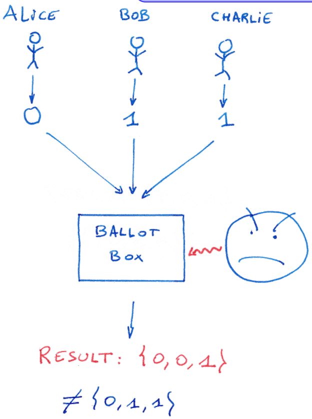
else return 1

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

Individual verifiability: game-b

As before:
Attacker has vote and cast oracles

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



$\underline{\mathsf{Exp}^{\mathsf{verif}}_{\mathcal{A}}(\lambda)}$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$

$\mathcal{A}^{\mathcal{O}_{\text{vote}}, \mathcal{O}_{\text{cast}}}(\mathsf{pk})$

$r \leftarrow \mathsf{Tally}(\mathsf{BB}, \mathsf{sk})$

if $\exists V_c.\ r = \rho(\mathsf{Voted} \cup V_c)$ **then**

  return 0

else return 1

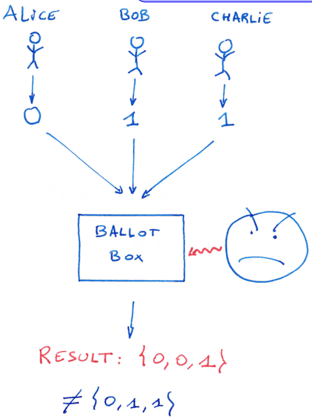
Advantage of the adversary: $\mathsf{P}\left[\mathsf{Exp}^{\mathsf{verif}}_{\mathcal{A}}(\lambda) = 1\right]$

Individual verifiability: game-b

As before:
Attacker has vote and cast oracles

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

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

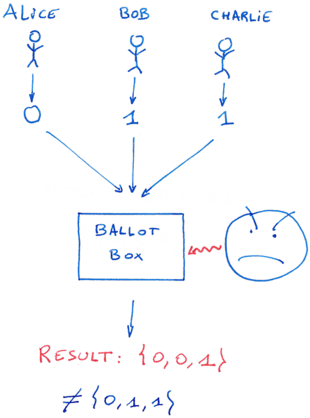


$\text{Exp}^{\text{verif}}_{\mathcal{A}}(\lambda)$

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

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

Individual verifiability: game-based definition



Compute the election result

$\mathsf{Exp}^{\mathsf{verif}}_{\mathcal{A}}(\lambda)$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$

$\mathcal{A}^{\mathcal{O}_{\mathsf{vote}}, \mathcal{O}_{\mathsf{cast}}}(\mathsf{pk})$

$r \leftarrow \mathsf{Tally}(\mathsf{BB}, \mathsf{sk})$

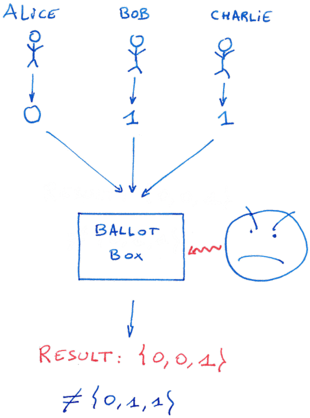
if $\exists V_c.\ r = \rho(\mathsf{Voted} \cup V_c)$ **then**

  return 0

else return 1

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

Individual verifiability: game-based definition



Compute the election result

$\mathsf{Exp}^{\mathsf{verif}}_{\mathcal{A}}(\lambda)$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$

$\mathcal{A}^{\mathcal{O}_{\mathsf{vote}}, \mathcal{O}_{\mathsf{cast}}}(\mathsf{pk})$

$r \leftarrow \mathsf{Tally}(\mathsf{BB}, \mathsf{sk})$

if $\exists V_c.\ r = \rho(\mathsf{Voted} \cup V_c)$ **then**

  return 0

else return 1

Result contains at least honest votes?
if not: $\mathcal{A}$ wins

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

Main result

Theorem (Privacy implies Individual Verifiability (computational))

$$\exists \mathcal{A}.\ \mathsf{P}\left[\mathsf{Exp}^{\mathsf{verif}}_{\mathcal{A}}(\lambda) = 1\right] \ \textit{not negligible} \quad \Longrightarrow$$

$$\exists \mathcal{B}.\ \left|\mathsf{P}\left[\mathsf{Exp}^{\mathsf{priv},0}_{\mathcal{B}}(\lambda) = 1\right] - \mathsf{P}\left[\mathsf{Exp}^{\mathsf{priv},1}_{\mathcal{B}}(\lambda) = 1\right]\right| \ \textit{not negligible.}$$

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

Theorem (Privacy implies Individual Verifiability (symbolic))

$$\forall \alpha, a, b.\ P_{\alpha \cup \{a \mapsto 0, b \mapsto 1\}} \approx P_{\alpha \cup \{a \mapsto 1, b \mapsto 0\}} \ \Longrightarrow$$

$$\forall \alpha.\ \forall (t.\mathtt{out}(ch_r, x), \phi) \in \mathsf{trace}(P_\alpha).\ \exists V_c.\ \phi(x) = \rho(\mathit{Voted}(t) \uplus V_c).$$

Intuition

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

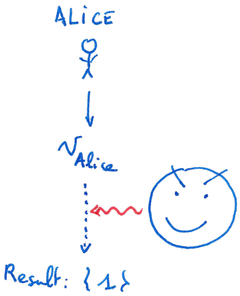
Intuition:

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



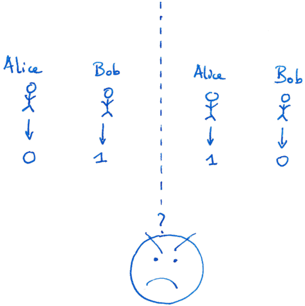
Intuition

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

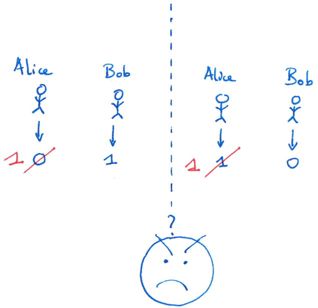


Intuition

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

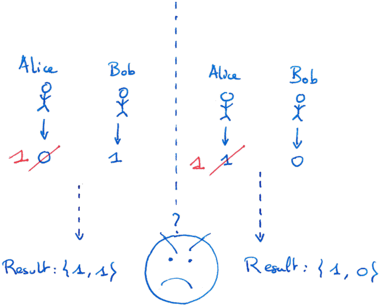


Intuition

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 $\{1, \text{Bob's vote}\}$

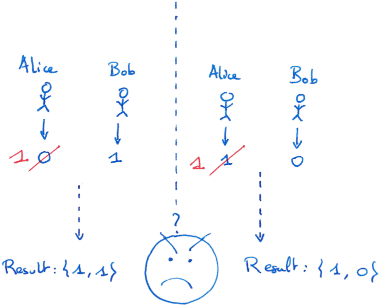


Intuition

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 $\{1, \text{Bob's vote}\}$



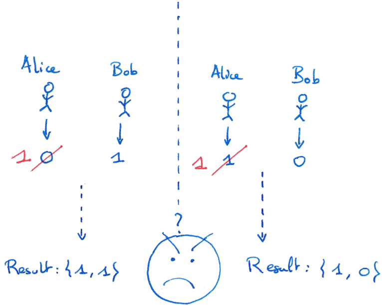
$\Longrightarrow$ the attacker learns Bob's vote, and breaks privacy

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}



$\Longrightarrow$ 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 $\mathcal{A}$ breaks verifiability we build $\mathcal{B}$ that breaks privacy.

$\mathcal{A}$

$\beta = 0$ | $\beta = 1$

$\mathcal{B}$

Proof sketch (assuming a blank vote)

Assuming $\mathcal{A}$ breaks verifiability we build $\mathcal{B}$ that breaks privacy.

$\mathcal{A}$ $id_1 \dots\dots\dots id_n$

$\underline{\beta = 0}$ $\underline{\beta = 1}$

$\mathcal{B}$

- Say $\mathcal{A}$ uses voters $id_1, \dots, id_n$.
  $\mathcal{B}$ will add n fresh voters: $id_1, \dots, id_n, id'_1, \dots, id'_n$.

Proof sketch (assuming a blank vote)

Assuming $\mathcal{A}$ breaks verifiability we build $\mathcal{B}$ that breaks privacy.

$\mathcal{A}$ $id_1 \ldots\ldots id_n$

$\underline{\beta = 0}$ $\underline{\beta = 1}$

$\mathcal{B}$ $id_1 \ldots id_n$ $id_1 \ldots id_n$

- Say $\mathcal{A}$ uses voters $id_1, \ldots, id_n$.
  $\mathcal{B}$ will add n fresh voters: $id_1, \ldots, id_n, id'_1, \ldots, id'_n$.

Proof sketch (assuming a blank vote)

Assuming $\mathcal{A}$ breaks verifiability we build $\mathcal{B}$ that breaks privacy.

$\mathcal{A}$ $id_1 \dots\dots id_n$

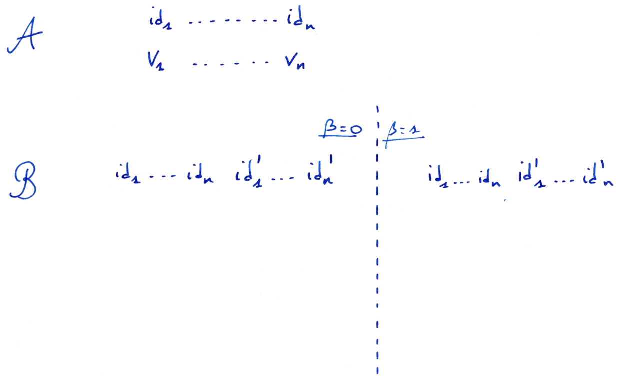
$\beta=0$ | $\beta=1$

$\mathcal{B}$ $id_1 \dots id_n \; id'_1 \dots id'_n$ | $id_1 \dots id_n \; id'_1 \dots id'_n$

- Say $\mathcal{A}$ uses voters $id_1, \dots, id_n$.
  $\mathcal{B}$ will add n fresh voters: $id_1, \dots, id_n, id'_1, \dots, id'_n$.
- $\mathcal{B}$ simulates $\mathcal{A}$.
  Whenever $\mathcal{A}$ makes id_i vote for v_i,
  $\mathcal{B}$ makes id_i vote v_i on the left, and *blank* on the right.

Proof sketch (assuming a blank vote)

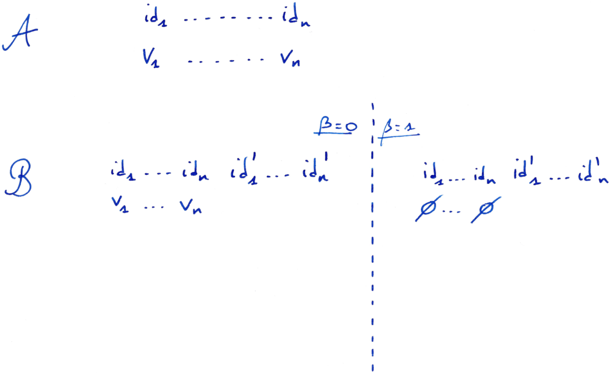
Assuming $\mathcal{A}$ breaks verifiability we build $\mathcal{B}$ that breaks privacy.



- Say $\mathcal{A}$ uses voters $id_1, \dots, id_n$.
  $\mathcal{B}$ will add n fresh voters: $id_1, \dots, id_n, id'_1, \dots, id'_n$.
- $\mathcal{B}$ simulates $\mathcal{A}$.
  Whenever $\mathcal{A}$ makes id_i vote for v_i,
  $\mathcal{B}$ makes id_i vote v_i on the left, and *blank* on the right.

Proof sketch (assuming a blank vote)

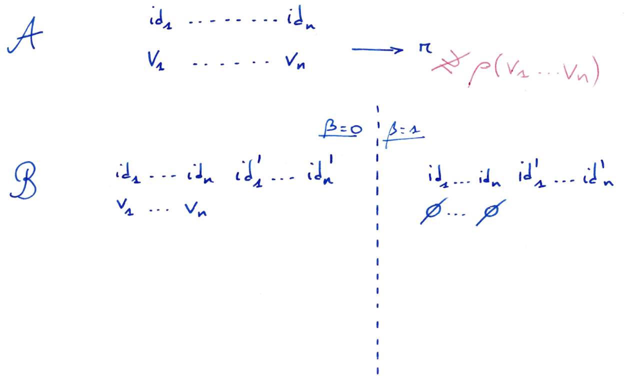
Assuming $\mathcal{A}$ breaks verifiability we build $\mathcal{B}$ that breaks privacy.



- Say $\mathcal{A}$ uses voters $id_1, \dots, id_n$.
  $\mathcal{B}$ will add n fresh voters: $id_1, \dots, id_n, id'_1, \dots, id'_n$.
- $\mathcal{B}$ simulates $\mathcal{A}$.
  Whenever $\mathcal{A}$ makes id_i vote for v_i,
  $\mathcal{B}$ makes id_i vote v_i on the left, and *blank* on the right.

Proof sketch (assuming a blank vote)

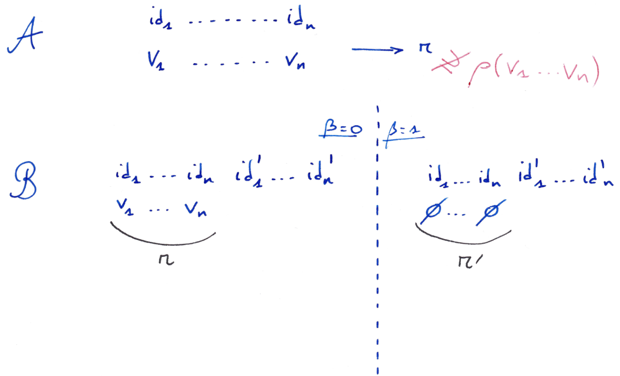
Assuming $\mathcal{A}$ breaks verifiability we build $\mathcal{B}$ that breaks privacy.



- 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)

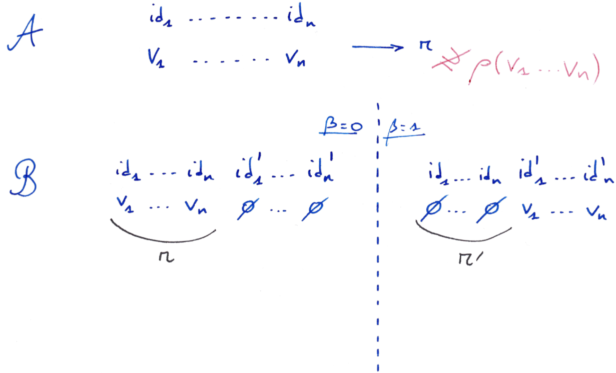
Assuming $\mathcal{A}$ breaks verifiability we build $\mathcal{B}$ that breaks privacy.



- 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)

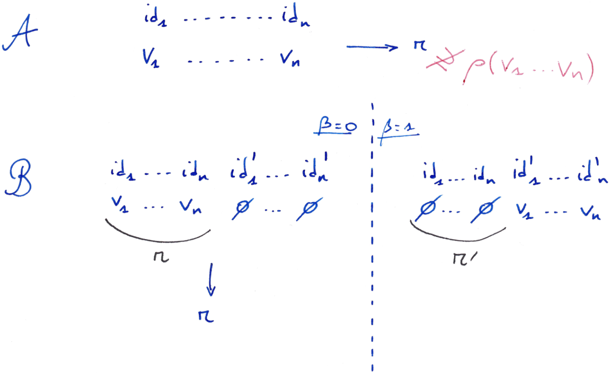
Assuming $\mathcal{A}$ breaks verifiability we build $\mathcal{B}$ that breaks privacy.



- 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'.
- $\mathcal{B}$ then makes each id_i vote *blank* on the left, and v_i on the right.

Proof sketch (assuming a blank vote)

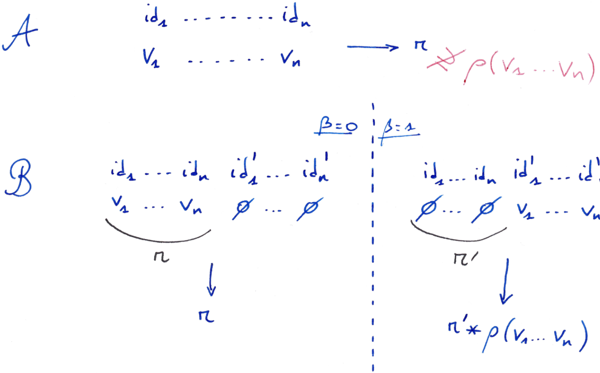
Assuming $\mathcal{A}$ breaks verifiability we build $\mathcal{B}$ that breaks privacy.



- The sets of honest votes are the same on both sides: $\mathcal{B}$ gets the result.
- The result is:
  - on the left: $r * blank^n = r$
  - on the right: $r' * v_1 * \ldots * v_n$

Proof sketch (assuming a blank vote)

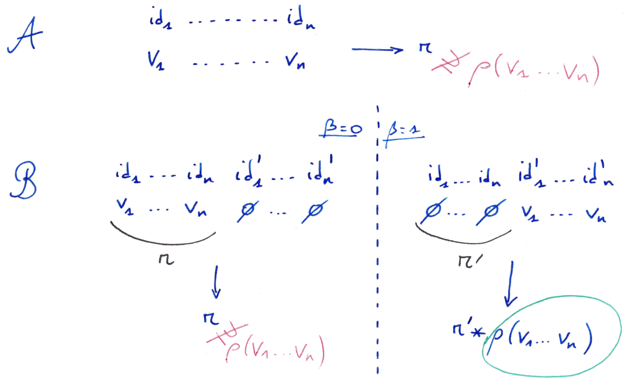
Assuming $\mathcal{A}$ breaks verifiability we build $\mathcal{B}$ that breaks privacy.



- The sets of honest votes are the same on both sides: $\mathcal{B}$ gets the result.
- The result is:
  - on the left: $r * blank^n = r$
  - on the right: $r' * v_1 * \ldots * v_n$

Proof sketch (assuming a blank vote)

Assuming $\mathcal{A}$ breaks verifiability we build $\mathcal{B}$ that breaks privacy.



- $\mathcal{B}$ 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 *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*

  $\rightarrow$ Our result:

  Privacy $\Rightarrow$ 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*

  $\rightarrow$ Our result:

  Privacy $\Rightarrow$ Individual verifiability *with the same trust assumptions*

  $\rightarrow$ 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
  $\longrightarrow$ considerably weakens privacy!

The problem with privacy

- Problem with existing game-based definitions:
  the ballot box is assumed honest
  $\longrightarrow$ 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
  $\longrightarrow$ The result is just Alice's vote!

The problem with privacy

- Problem with existing game-based definitions:
  the ballot box is assumed honest
  $\longrightarrow$ 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
  $\longrightarrow$ 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
  $\Longrightarrow$ let's introduce the verification steps of the protocol in privacy!

Our proposition: privacy with careful voters

- Privacy *is* linked with verifiability
  $\Longrightarrow$ 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:

$\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv-careful},\beta}(\lambda)$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$

$\mathsf{BB} \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{vote}}}(\mathsf{pk})$

$\mathcal{A}_2^{\mathcal{O}^{\mathsf{BB}}_{\mathsf{happy}}}(\mathsf{pk})$

if $\forall id \in \mathsf{V}_0, \mathsf{V}_1.\ id \in \mathsf{H} \ \wedge\ \rho(\mathsf{V}_0) = \rho(\mathsf{V}_1)$ **then**

$\quad r \leftarrow \mathsf{Tally}(\mathsf{BB}, \mathsf{sk})$

else $r \leftarrow \perp$

return $\mathcal{A}_3(\mathsf{pk}, r)$

$\mathcal{O}_{\mathsf{vote}}(id, v_0, v_1)$

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

$\mathsf{V}_i \leftarrow \mathsf{V}_i \| v_i$ for $i \in \{0, 1\}$

$\mathsf{L}_{id} \leftarrow \mathsf{L}_{id} \| (b, v_\beta)$

return b

$\mathcal{O}^{\mathsf{BB}}_{\mathsf{happy}}(id)$

if $\mathsf{VerifVoter}(id, \mathsf{L}_{id}, \mathsf{BB})$ **then**

$\quad \mathsf{H} \leftarrow \mathsf{H} \| id$

Our proposition: privacy with careful voters

- Privacy *is* lin Dishonest ballot box: provided by the attacker
  $\Longrightarrow$ 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:

$\underline{\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv-careful},\beta}(\lambda)}$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$

$\mathsf{BB} \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{vote}}}(\mathsf{pk})$

$\mathcal{A}_2^{\mathcal{O}_{\mathsf{happy}}^{\mathsf{BB}}}(\mathsf{pk})$

if $\forall id \in \mathsf{V}_0, \mathsf{V}_1.\ id \in \mathsf{H} \ \wedge\ \rho(\mathsf{V}_0) = \rho(\mathsf{V}_1)$ **then**

$\quad r \leftarrow \mathsf{Tally}(\mathsf{BB}, \mathsf{sk})$

else $r \leftarrow \perp$

return $\mathcal{A}_3(\mathsf{pk}, r)$

$\underline{\mathcal{O}_{\mathsf{vote}}(id, v_0, v_1)}$

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

$\mathsf{V}_i \leftarrow \mathsf{V}_i \| v_i$ for $i \in \{0, 1\}$

$\mathsf{L}_{id} \leftarrow \mathsf{L}_{id} \| (b, v_\beta)$

return b

$\underline{\mathcal{O}_{\mathsf{happy}}^{\mathsf{BB}}(id)}$

if $\mathsf{VerifVoter}(id, \mathsf{L}_{id}, \mathsf{BB})$ **then**

$\quad \mathsf{H} \leftarrow \mathsf{H} \| id$

Our proposition: privacy with careful voters

- Privacy *is* lin Dishonest ballot box: provided by the attacker

  $\Longrightarrow$ let's introduce the verification steps of the protocol in privacy!

  Vote oracle as before

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

$\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv-careful},\beta}(\lambda)$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^{\lambda})$

$\mathsf{BB} \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{vote}}}(\mathsf{pk})$

$\mathcal{A}_2^{\mathcal{O}_{\mathsf{happy}}^{\mathsf{BB}}}(\mathsf{pk})$

if $\forall id \in \mathsf{V}_0, \mathsf{V}_1.\ id \in \mathsf{H}\ \wedge\ \rho(\mathsf{V}_0) = \rho(\mathsf{V}_1)$ **then**

  $r \leftarrow \mathsf{Tally}(\mathsf{BB}, \mathsf{sk})$

else $r \leftarrow \perp$

return $\mathcal{A}_3(\mathsf{pk}, r)$

$\mathcal{O}_{\mathsf{vote}}(id, v_0, v_1)$

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

$\mathsf{V}_i \leftarrow \mathsf{V}_i \| v_i$ for $i \in \{0, 1\}$

$\mathsf{L}_{id} \leftarrow \mathsf{L}_{id} \| (b, v_{\beta})$

return b

$\mathcal{O}_{\mathsf{happy}}^{\mathsf{BB}}(id)$

if $\mathsf{VerifVoter}(id, \mathsf{L}_{id}, \mathsf{BB})$ **then**

  $\mathsf{H} \leftarrow \mathsf{H} \| id$

Our proposition: privacy with careful voters

- Privacy *is* linked with verifiability
  $\Longrightarrow$ let's introduce the verification steps of the protocol in privacy!

$\mathcal{A}$ triggers voters' verifications

- The attacker c
  provided all voters perform the verifications:

$\mathrm{Exp}_{\mathcal{A}}^{\mathrm{priv-careful},\beta}(\lambda)$

$(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathrm{Setup}(1^{\lambda})$

$\mathrm{BB} \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathrm{vote}}}(\mathrm{pk})$

$\mathcal{A}_2^{\mathcal{O}_{\mathrm{happy}}^{\mathrm{BB}}}(\mathrm{pk})$

if $\forall id \in \mathrm{V}_0, \mathrm{V}_1.\ id \in \mathrm{H}\ \wedge\ \rho(\mathrm{V}_0) = \rho(\mathrm{V}_1)$ **then**

  $r \leftarrow \mathrm{Tally}(\mathrm{BB}, \mathrm{sk})$

else $r \leftarrow \perp$

return $\mathcal{A}_3(\mathrm{pk}, r)$

$\mathcal{O}_{\mathrm{vote}}(id, v_0, v_1)$

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

$\mathrm{V}_i \leftarrow \mathrm{V}_i \| v_i$ for $i \in \{0, 1\}$

$\mathrm{L}_{id} \leftarrow \mathrm{L}_{id} \| (b, v_\beta)$

return b

$\mathcal{O}_{\mathrm{happy}}^{\mathrm{BB}}(id)$

if $\mathrm{VerifVoter}(id, \mathrm{L}_{id}, \mathrm{BB})$ **then**

  $\mathrm{H} \leftarrow \mathrm{H} \| id$

Our proposition: privacy with careful voters

- Privacy *is* linked with verifiability
  $\Longrightarrow$ 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:

$\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv-careful},\beta}(\lambda)$

Tally only if all voters have verified

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$

$\mathsf{BB} \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{vote}}}(\mathsf{pk})$

$\mathcal{A}_2^{\mathcal{O}_{\mathsf{happy}}^{\mathsf{BB}}}(\mathsf{pk})$

$\textbf{if } \forall id \in \mathsf{V}_0, \mathsf{V}_1.\ id \in \mathsf{H} \ \wedge \ \rho(\mathsf{V}_0) = \rho(\mathsf{V}_1) \textbf{ then}$

$\quad r \leftarrow \mathsf{Tally}(\mathsf{BB}, \mathsf{sk})$

$\textbf{else } r \leftarrow \perp$

$\textbf{return } \mathcal{A}_3(\mathsf{pk}, r)$

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

$\mathsf{V}_i \leftarrow \mathsf{V}_i \| v_i \text{ for } i \in \{0, 1\}$

$\mathsf{L}_{id} \leftarrow \mathsf{L}_{id} \| (b, v_\beta)$

$\textbf{return } b$

$\mathcal{O}_{\mathsf{happy}}^{\mathsf{BB}}(id)$

$\textbf{if } \mathsf{VerifVoter}(id, \mathsf{L}_{id}, \mathsf{BB}) \textbf{ then}$

$\quad \mathsf{H} \leftarrow \mathsf{H} \| id$

Our proposition: privacy with careful voters

- Our result still holds for our new definition:

Theorem

Privacy against a dishonest ballot box with careful voters $\Longrightarrow$

Individual Verifiability against a dishonest ballot box

Our proposition: privacy with careful voters

- Our result still holds for our new definition:

Theorem

Privacy against a dishonest ballot box with careful voters $\Longrightarrow$

Individual Verifiability against a dishonest ballot box

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

Protocol	Honest box	Dishonest box naïve	Careful voters	
Helios	✓	✗	✗	[attack P. Roenne]
Belenios	✓	✗	✓	
Civitas (no revote)	✓	✗	✓	
Neuchâtel (no revote)	✓	✗	✗	[assumes an honest box]

✓: the protocol is private, ✗: attack on privacy

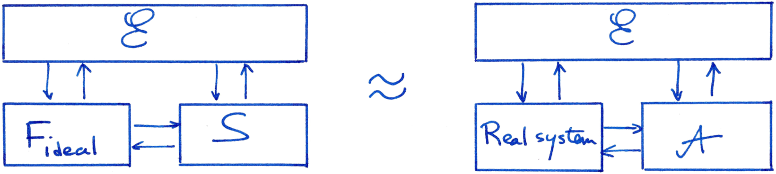
Work in progress: towards more precise definitions

- 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.

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

Conclusion

- A counter-intuitive result:

$$\text{Privacy} \Longrightarrow \text{Individual Verifiability}$$

- Proved in computational and symbolic models
- Better understanding of privacy: some verifiability is required!

Conclusion

- A counter-intuitive result:

$$\text{Privacy} \Longrightarrow \text{Individual Verifiability}$$

- 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:

$$\text{Privacy} \implies \text{Individual Verifiability}$$

- 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]
- A new definition of privacy against a dishonest ballot box
  $\longrightarrow$ modelling verification steps
- Limitation: assumes everyone checks their vote
  $\longrightarrow$ Future work: more plausible scenario where only some voters check