Ballot privacy in elections: new metrics and constructions.

Olivier Pereira – Université catholique de Louvain

Based on joint works with:
D. Bernhard, V. Cortier, E. Cuvelier,
T. Peters and B. Warinschi

March 2015
Open Voting
Open Voting
Open Voting

- Every voter can verify that nobody tampered with her/his vote
- Every voter can compute the tally
- No privacy, no coercion-resistance, no fairness, ...
Secret Ballot

- Liberal motivation: “My vote is my own business, elections are a tool for aggregating private opinions”
- Practical motivation: Prevent coercion and bribery
A traditional paper approach

- With voting booth: privacy, coercion-resistance, fairness, ...
- If a voter keeps an eye on the urn and tally all day long, he can be convinced that:
  - his vote is untampered
  - the tally is based on valid votes and correct
- A minute of inattention is enough to break this
Privacy vs Verifiability – Two Extremes

Hand raising vote

Verifiability 100%
Privacy 0%

Uncontrolled ballot box

Verifiability 0%
Privacy 100%
Privacy and Verifiability
Defining Vote Privacy

Not an absolute notion:

- Usually accepted that there is no privacy when all voters support the same candidate

Elections as Secure Function Evaluation [Yao82]:

- “The voting system should not leak more than the outcome”
- But we would like to know how much the outcome leaks!

Game-style definition [KTV11]:

- Privacy measured as max probability to distinguish whether I voted in one way or another
- Often too strong: that probability is \( \approx 1 \) when:

\[
\#\text{different ballots} \gg \#\text{voters}
\]
Defining Vote Privacy

What do we want to measure?

1. With what probability can $A$ guess my vote? 
   Sounds like min-entropy!

2. In how many ways can I pretend that I voted? 
   Sounds like Hartley entropy!
Let:

- $\mathcal{D}$ be the distribution of honest votes (if known)
- $T : \sup(\mathcal{D}) \rightarrow \{0, 1\}^*$ be a target function
  - $T(v_1, \ldots, v_n) := v_i$
  - $T(v_1, \ldots, v_n) := (v_i \neq v_j)$
- $\rho(v_1, \ldots, v_n)$ be the official outcome of the election
- $\text{view}_A(\mathcal{D}, \pi)$ be the view of $A$ participating to voting protocol $\pi$ in which honest voters vote according to $\mathcal{D}$
Measure(s) for privacy

\[ M_x(T, D, \pi) := \inf_{\mathcal{A}} F_x(T(D)|\text{view}_A(D, \pi), \rho(D, v_A)) \]

where:
- \( F_x(A|B) \) is some x-Rényi entropy measure on \( A \) given \( B \)
Choices for \( F_X(A|B) \)

\[
M_X(T, D, \pi) := \inf_A F_X(T(D)|\text{view}_A(D, \pi), \rho(D, v_A))
\]

Choices for \( F_X(A|B) \):

\( \tilde{H}_\infty \) Average min-entropy: \(-\log \left( \mathbb{E}_{b \in B} \left[ 2^{-H_\infty(A|B=b)} \right] \right) \) [DORS08]

Measures the probability that \( A \) guesses the target

\( H_{\perp \infty} \) Min-min-entropy: \( \min_{b \in B} H_\infty(A|B = b) \)

Same as before, but for the worst possible \( b \)

\( H_{\perp 0} \) Min-Hartley-entropy: \( \min_{b \in B} H_0(A|B = b) \)

Measures the number of values that the target can take for the worst \( b \) – No probabilities involved!
An example...

Consider:

- An approval (yes/no) election with 1 question
- 3 voters voting uniformly at random
- target is the first voter

\[
\begin{array}{|c|c|c|}
\hline
\rho_1 := \bot & H_\infty & H_\infty^\bot & H_0^\bot \\
\hline
\rho_2 := |\vec{v}|_{yes} > |\vec{v}|_{no} & .4 & .4 & 1 \\
\rho_3 := (|\vec{v}|_{yes}, |\vec{v}|_{no}) & .4 & 0 & 0 \\
\rho_4 := \vec{v} & 0 & 0 & 0 \\
\hline
\end{array}
\]

(.4 \approx - \log \frac{3}{4})
Scantegrity Audit Data

- Official outcome: number of votes received by each candidate
- Scantegrity audit trail exposes all ballots (codes removed)
- Scantegrity take-home receipt shows how many bullets you filled
**Scantegrity Audit Data**

From the 2009 Takoma Park municipal election data:

<table>
<thead>
<tr>
<th>Ward</th>
<th>1</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Ballots</td>
<td>470</td>
<td>85</td>
<td>198</td>
</tr>
<tr>
<td>Question</td>
<td>A</td>
<td>B</td>
<td>A</td>
</tr>
<tr>
<td>$H_0^\perp$ from official outcome</td>
<td>6</td>
<td>3.17</td>
<td>6</td>
</tr>
<tr>
<td>$H_0^\perp$ with receipts</td>
<td>1.58</td>
<td>1.58</td>
<td>0</td>
</tr>
</tbody>
</table>

- 6/3.17 bits is a question with 3/2 candidates to rank (including incorrect rankings)
- In most cases, rankings of a certain length are uncommon
- In Ward 5, a voter looses his/her privacy completely on Question A if he/she shows his/her receipt!
Single-Pass Cryptographic Voting

A common approach ([CGS97], [DJ01], Helios, . . .):

1. Trustees create an election public key $pk$
2. Voters publish an encryption of their vote $v_i$
3. Trustees compute and publish the tally, using the secret key $sk$
4. Everyone can verify that the tally is consistent with the encrypted votes
Cryptographic Voting

Problem with entropic measures of privacy:

\[ H(v_i | Enc_{pk}(v_i), pk) = 0 \]

Solution: use a computational analog of entropy:

\[ F_c^x(A|B) \geq r \iff \exists B' \approx^c B \text{ and } F_x(A|B') \geq r \]

In particular,

\[ H^c(v_i | Enc_{pk}(v_i), pk) \geq r \quad \text{if} \quad H(v_i | Enc_{pk}(0), pk) \geq r \]
Computational Measure(s) for privacy

\[ M^c_x(T, D, \pi) := \inf_{A} F^c_x(T(D)|\text{view}_A(D, \pi), \rho(D, v_A)) \]

where:

- \( F^c_x(A|B) \) is a \( \chi \)-Rényi computational entropy metric on \( A \) given \( B \)

**Definition** (informal): A voting scheme \( \pi \) with tallying function \( \rho \) offers *ballot privacy* if, for all \( T, D \):

\[ M^c_x(T, D, \pi) = \inf_{A} F^c_x(T(D)|\rho(D, v_A)) \]
Privacy and Verifiability

Do we *need* to move to computational entropies?

- Publish encrypted votes, but what if encryption gets broken?
  - because time passes and computing speed increases
  - because decryption keys are lost/stolen
  - because there is an algorithmic breakthrough
Voting with a Perfectly Private Audit Trail

Can we offer verifiability without impacting privacy?

More precisely:
Can we take a non-verifiable voting scheme and add verifiability without impacting privacy?

Goal:

▶ Have a new kind of audit data
▶ Audit data must perfectly hide the votes
▶ Usability must be preserved:
  1. Practical distributed key generation
  2. No substantial increase of the cost of ballot preparation
  3. Be compatible with efficient proof systems
Commitments Can Enable Perfect Privacy

- A commitment is *perfectly hiding* if $d$ is independent of $m$
- A commitment is *computationally binding* if it is *infeasible* to produce $d, (m, a), (m', a')$ such that $d$ can be opened on both $(m, a)$ and $(m', a')$ ($m \neq m'$)

Example:
- Let $g_0, g_1$ be random generators of a cyclic group $G$
- Set $d = g_0^a g_1^m$ as a commitment on $m$ with random opening $a$
- Finding a different $(m, a)$ pair consistent with $d$ is as hard as computing the discrete log of $g_1$ in base $g_0$
A New Primitive: Commitment Consistent Encryption

Commitment Consistent Encryption (CCE) scheme
\[ \Pi = (\text{Gen}, \text{Enc}, \text{Dec}, \text{DerivCom}, \text{Open}, \text{Verify}) \]

\((\text{Gen}, \text{Enc}, \text{Dec})\) is a classic encryption scheme
\[ c = \text{Enc}_{pk}(m) \]

\(\text{DerivCom}_{pk}(c)\) from the ciphertext, derives a commitment \(d\)
\(\text{Open}_{sk}(c)\) outputs an opening value \(a\) from \(c\) using \(sk\)
\(\text{Verify}_{pk}(d, a, m)\) checks that \(d\) is a commitment on \(m\) w.r.t. \(a\)
Single-Pass Cryptographic Voting

Voting with a CCE scheme:

1. Trustees create an election public key $pk$
2. Voters submit an encryption of their vote $v_i$ to Trustees
3. Trustees publish commitments extracted from encrypted votes
4. Trustees publish the tally, as well a proofs of correctness
Voting with a Perfectly Private Audit Trail

If:

- Commitments are perfectly hiding
- Proofs are perfect/statistical zero-knowledge

Then:

- the audit trail is independent of the votes

\[ H_x(\text{votes} \mid \text{audit trail + tally}) = H_x(\text{votes} \mid \text{tally}) \]

If cryptographic assumptions are broken:

- Someone might be able to “prove” a wrong result

But:

- Proof needs to be produced fast enough to be compelling
- Only people who believe in crypto assumption will trust the proof
Building CC Encryption Schemes

Group setup:

$\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ different groups of same prime order

A bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$

<table>
<thead>
<tr>
<th>$\mathbb{G}_1$</th>
<th>$\mathbb{G}_2$</th>
<th>$\mathbb{G}_T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g$</td>
<td>$h$</td>
<td>$e(g, h)$</td>
</tr>
<tr>
<td>$g^a$</td>
<td>$h$</td>
<td>$e(g^a, h) = e(g, h)^a$</td>
</tr>
<tr>
<td>$g$</td>
<td>$h^b$</td>
<td>$e(g, h^b) = e(g, h)^b$</td>
</tr>
</tbody>
</table>

DDH problem expected to be hard in $\mathbb{G}_1$ and $\mathbb{G}_2$
The PPATS Scheme

Additively homomorphic scheme for small message $m \in \mathbb{Z}_q$

<table>
<thead>
<tr>
<th>$\mathbb{G}_1$</th>
<th>$\mathbb{G}_2$</th>
<th>$\mathbb{G}_T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g, g_1 = g^{x_1}$</td>
<td>$h, h_1$</td>
<td>$d = h^r h_1^m$</td>
</tr>
<tr>
<td>$c_1 = g^s$</td>
<td></td>
<td>$Dec_{sk}(c) : DLog$ of</td>
</tr>
<tr>
<td>$c_2 = g^r g_1^s$</td>
<td>$d = h^r h_1^m$</td>
<td>$e(c_1^{x_1}/c_2, h)$</td>
</tr>
<tr>
<td>$Open_{sk}(c) :$</td>
<td>$a = c_2/c_1^{x_1}$</td>
<td>$e(g, d)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$= e(g, h_1)^m$</td>
</tr>
<tr>
<td>$Verif_{pk}(d, m, a) :$</td>
<td></td>
<td>$e(a, h)$ $\overset{?}{=} e(g, d/h_1^m)$</td>
</tr>
</tbody>
</table>
Efficiency Comparisons

Assuming:

- 256 bit multiplication costs 1
- multiplication has quadratic complexity
- exponentiation/point multiplication by square and multiply

Cost of 1 encryption (+ 0/1 proof)

<table>
<thead>
<tr>
<th>Scheme</th>
<th>$\mathbb{Z}^*_p$</th>
<th>$\mathbb{Z}^*_N^2$</th>
<th>$G_1$</th>
<th>$G_2$</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pedersen/Paillier</td>
<td>4</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>8.650.752</td>
</tr>
<tr>
<td>PPATS</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>6</td>
<td>115.200</td>
</tr>
</tbody>
</table>

+ PPATS has considerably simpler threshold variants, thanks to the public order groups
Conclusions: Privacy and Verifiability

Two apparently conflicting requirements on votes:

Hiding for privacy ↔ Showing for verifiability

Commitment-consistent encryption can reconcile these goals!

Experiences and metrics are useful: the outcome of an election can, in itself, give more information than expected, as voters vote highly non-uniformly!