How to fake zero-knowledge proofs, again

Véronique Cortier[1] Pierrick Gaudry[1] Quentin Yang[12]

[1]LORIA - CNRS, INRIA Nancy - Grand-Est, Université de Lorraine

[2]École Polytechnique, Palaiseau

October 2020

Zero Knowledge Proofs in Electronic Voting

Proving the validity of the ballot is necessary, and often requires zero knowledge proofs.

Let's consider an election where you can choose between two candidates :

To vote Alice

Alice	Bob
1	0

To vote Bob

Alice	Bob
0	1

To vote Blank

Alice	Bob
0	0

Zero Knowledge Proofs in Electronic Voting

Proving the validity of the ballot is necessary, and often requires zero knowledge proofs.

Let's consider an election where you can choose between two candidates :

To vote Alice

Alice	Bob
1	0

To vote Bob

Alice	Bob
0	1

To vote Blank

Alice	Bob
0	0

Alice	Bob
-23	42

Zero Knowledge Proofs in Electronic Voting

Proving the validity of the ballot is necessary, and often requires zero knowledge proofs.

Let's consider an election where you can choose between two candidates :

To vote Alice

Alice	Bob
1	0

To vote Bob

Alice	Bob
0	1

To vote Blank

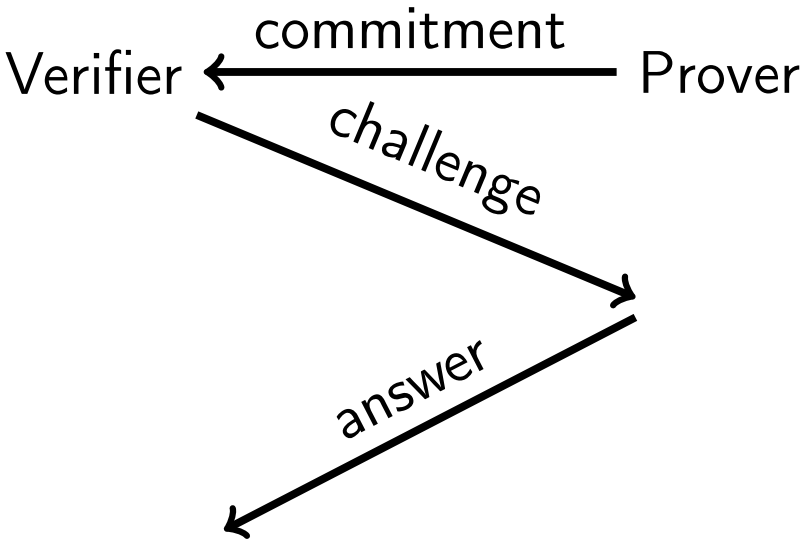
Alice	Bob
0	0

Not allowed

Alice	Bob
−23	42

Zero Knowledge Proofs in Electronic Voting

Zero-knowledge proofs usually require an **interactive** protocol :



Zero Knowledge Proofs in Electronic Voting

Zero-knowledge proofs usually require an **interactive** protocol, which is turned into a **non-interactive** one thanks to the Fiat-Shamir heuristic :

$$d = \mathrm{hash}(c),$$

where d is the challenge and
c is the commitment.

Zero Knowledge Proofs in Electronic Voting

Zero-knowledge proofs usually require an **interactive** protocol, which is turned into a **non-interactive** one thanks to the Fiat-Shamir heuristic :

$$d = \mathrm{hash}(c),$$

where d is the challenge and
c is the commitment.

However, this heuristic is too weak. (David Bernhard, Olivier Pereira, and Bogdan Warinschi, ASIACRYPT 2012)

Zero Knowledge Proofs in Belenios

Belenios is a verifiable online voting system.
It uses ElGamal encryption with public key (g, h) and secret key s such that $h = g^s$.
The challenge is chosen with the operation

$$d = \text{hash}(\boldsymbol{x}, c),$$

where d is the challenge,
x is the cyphertext and
c is the commitment.
This is **not** the weak Fiat-Shamir transformation

Zero Knowledge Proofs in Belenios

Belenios is a verifiable online voting system.
It uses ElGamal encryption with public key (g, h) and
secret key s such that $h = g^s$.
The challenge is chosen with the operation

$$d = \text{hash}(\boldsymbol{x}, c),$$

where d is the challenge,
x is the cyphertext and
c is the commitment.
This is **not** the weak Fiat-Shamir transformation

BUT

g and h do not take part into the equation,

which gives two additionnal degrees of freedom.

Zero Knowledge Proofs in Belenios

Assuming that they can corrupt both

- **the server** (so they can choose g, the public generator of G)
- and **the authorities** (so they can choose h, the public encrypting key),

the attackers can forge a cyphertext x

which decrypts into a chosen value m

while providing a proof π that x is an encryption of either 0 or 1.

How to submit a fake ballot in Belenios

In Belenios, **several** Zero Knowledge proofs are required for a valid ballot.

Alice	Bob	Charly	Diane
x_1	x_2	x_3	x_4
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
π_1	π_2	π_3	π_4

$\rightsquigarrow \quad \sum x_i \quad \longrightarrow \quad \pi_0$

How to submit a fake ballot in Belenios

In Belenios, **several** Zero Knowledge proofs are required for a valid ballot.

Alice	Bob	Charly	Diane
x_1	x_2	x_3	x_4
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
π_1	π_2	π_3	π_4

$\rightsquigarrow \quad \sum x_i \quad \longrightarrow \quad \pi_0$

You can use fake proof π to fool π_1 but...

That means you've already fixed g and h, the public keys.

How to submit a fake ballot in Belenios

In Belenios, **several** Zero Knowledge proofs are required for a valid ballot.

Alice	Bob	Charly	Diane
x_1	x_2	x_3	x_4
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
π_1	π_2	π_3	π_4

$\rightsquigarrow \quad \sum x_i \quad \longrightarrow \quad \pi_0$

You can use fake proof π to fool π_1 but...

That means you've already fixed g and h, the public keys.

An additionnal proof π_0 is required to forge a valid ballot.

But since π_0 is also a 0/1 proof so you can use the fake proof π again !

Summing up

Conditions for the attack on Belenios :

- An instance where **you can chose at most one candidate**.
- The server and the authorities are corrupted.

Summing up

Conditions for the attack on Belenios :

- An instance where **you can chose at most one candidate**.
- The server and the authorities are corrupted.

Nature of the attack :

The attackers can forge a valid ballot which gives a **chosen** number of voices to a chosen candidate.

Summing up

Conditions for the attack on Belenios :

- An instance where **you can chose at most one candidate**.
- The server and the authorities are corrupted.

Nature of the attack :

The attackers can forge a valid ballot which gives a **chosen** number of voices to a chosen candidate.

Consequence :

- The attacker can break verifiability (in addition to confidentiality).
- They can also plan an untraceable denial of service attack.

Conclusion

Using the Fiat-Shamir transformation is tricky :

Make sure to hash **all** available context!

Thank you !