# Belenios: a simple private and verifiable electronic voting system

Authors : Véronique Cortier, Pierrick Gaudry, Stephane Glondu

CNRS, Inria, Univ. Lorraine, France

Arnaud Labourel (AMU, CNRS, LIS)

Distributed team
DALGOrithms seminar

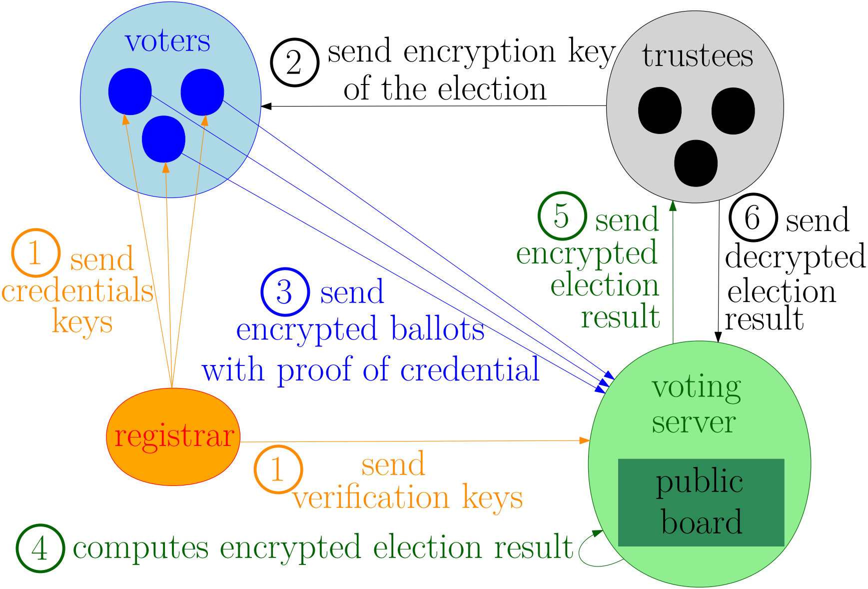
# Introduction

# Part II: the cryptographic tools

Previously, on the DALGO seminar, we have seen:

- the electronic voting system Belenios
- the global ideas behind it
- the properties guaranteed by the protocol: privacy and verifiability

# Global ideas behind Belenios



# Cryptographic tools used by Belenios

- Partially homomorphic ElGamal encryption scheme
  used to encrypt the votes
- Cryptographic hash function
  used for signatures and ZKP
- Non-interactive Zero-Knowledge Proofs (ZPK) used
  - by voters to prove validity of votes
  - by trustees to prove the correct decryption of the election result
- Schnorr signature scheme
  used to sign the ballot to prove the legitimacy of the vote
- Pedersen's threshold secret sharing scheme
  used by trustees s.t. no single authority has the private key of the election

# ElGamal encryption scheme

# ElGamal encryption scheme

- Asymmetric key encryption algorithm :
  - key $e$ to encode
  - key $d$ to decode
  - computationally intractable to decrypt encrypted an encrypted message without knowing $d$
- Created in 1985 by ElGamal
- Based on cyclic groups
- Security based on the difficulty of solving discrete logarithm in the chosen group

# Group

## Definition of a group

A group $(G, *)$ is a pair composed of a set $G$ and an operation $* : G \times G \to G$ s.t.:

- $*$ is associative (useful for fast exponentiation)
- there an identity element $i : \forall x, i * x = x = x * i$ (useful for defining an inverse)
- every element $x$ has an inverse $x^{-1}$ s.t. $x * x^{-1} = i$ (useful for defining $x/y = x * y^{-1}$)

For $x \in S$ and $y \in \mathbb{N}$, we define:

$$x^y = \underbrace{x * x * \cdots * x}_{y \text{ times}}$$

# Cyclic group

## Definition of a cyclic group

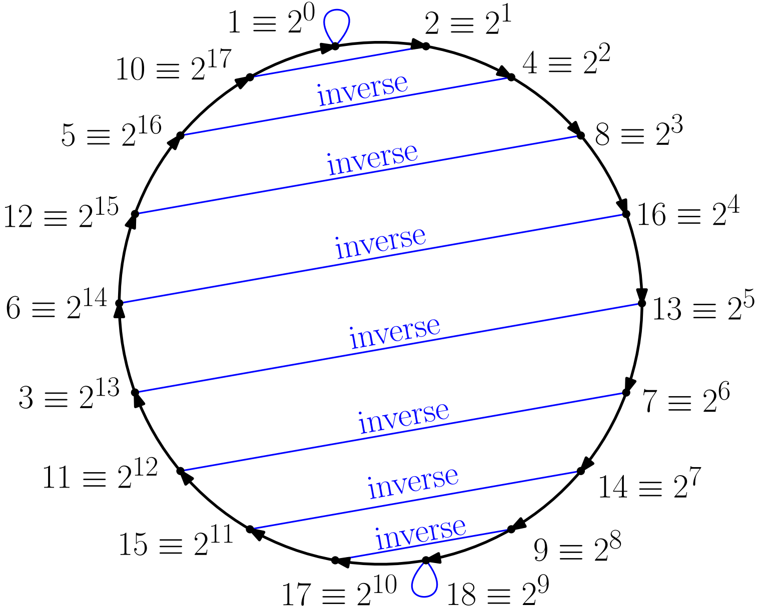
A group $(G, *)$ of order $q$ for which there is a generator $g$, i.e., an element of the group s.t.:

$$G = \{i = g^0, g^1, g^2, \ldots, g^{q-1}\}$$

**Examples of cyclic groups:**

- integers modulo $n$ with multiplication for prime $n$
- points of an elliptic curve on a finite field of prime order

# Group of integers $\pmod{19}$ with $g = 2$



# ElGamal encryption scheme

## Public information (part of the public key)

$(G, q, g)$ with :

- $G$ a cyclic group $G$ of order $q$
- $g$ a generator of $G$

## Decryption private key $d$

An integer $d$ randomly chosen from $\{1, \ldots, q-1\}$.

## Encryption public key $e$

$e = g^d$

# ElGamal encryption scheme

## Encryption of a vote $v \in \{0, 1\}$

1. Choose a random integer $r$ from $\{1, \ldots, q-1\}$
2. $\mathrm{enc}_e(v, r) = (g^r, e^r g^v)$

## Decryption of a message $(a, b)$

1. Compute $b/a^d = e^r g^v / (g^r)^d = (g^d)^r g^v / g^{rd} = g^v$
2. Compute $v = \mathrm{dec}_d(a, b)$ ($v = 1$ if $b/a^d = g$ and $0$ otherwise).

$(a^d)^{-1}$ can be computed as $a^{q-d}$ since
$(a^d) * a^{q-d} = g^{rd} * g^{r(q-d)} = g^{rq} = (g^q)^r = i^r = i$

# Homomorphic property of ElGamal encryption scheme

Recall that $\mathrm{enc}_e(v, r) = (g^r, e^r g^v)$.
We have :

$$\begin{aligned}
\mathrm{enc}_e\left(\sum_{i=1}^{n} v_i, \sum_{i=1}^{n} r_i\right) &= (g^{\sum_{i=1}^{n} r_i}, e^{\sum_{i=1}^{n} r_i} g^{\sum_{i=1}^{n} v_i}) \\
&= \left(\prod_{i=1}^{n} g^{r_i}, \prod_{i=1}^{n} e^{r_i} g^{v_i}\right) \\
&= \prod_{i=1}^{n} \mathrm{enc}_e(v_i, r_i)
\end{aligned}$$

# Assumptions for security of ElGamal

Some assumptions must hold on the chosen cyclic group $G$ of order $q$ to achieve security :

**Computational Diffie–Hellman assumption (CDH)**

Given $(g, g^a, g^b)$ for a randomly chosen generator $g$ of $G$ and random $a, b \in \{0, \ldots, q-1\}$, it is computationally intractable to compute the value $g^{ab}$

If CDH holds, then the encryption function is one-way (computationally intractable to decrypt encrypted messages without the decryption key)

# Assumptions for security of El Gamal

## Discrete logarithm assumption

Given $a$ and $b$, it is computationally intractable to compute the value $x$ s.t. $a^x = b$.

If computing the discrete logarithm in $G$ is easy, then the CDH problem could be solved easily:
Given $(g, g^a, g^b)$ :

- compute $a$ from $g$ and $g^a$
- compute $g^{ab} = (g^b)^a$

It is not known if this is the only method and so if the discrete log assumption is equivalent to the CDH assumption.

# Assumptions for security of ElGamal

Best known algorithms for discrete logarithm are super-polynomial in the size of the input (with **classic model** of computation).

This is not true for quantum computers:
variant of Shor's algorithm with polynomial (in the size of input) complexity.
$\Rightarrow$ ElGamal scheme is not quantum-resistant

# Assumptions for security of ElGamal

The fact that the encryption function is one-way does not imply that an adversary cannot learn information on the content of the encrypted messages.
PPTA: Probabilistic, Polynomial-Time Algorithm

## Semantic security

Any PPTA that is given $\mathrm{enc}(m)$, and $|m|$, cannot determine any partial information on $m$ with probability non-negligibly higher than all other PPTA's that only have access to $|m|$.

For semantic security, a stronger assumption than CDH is needed.

# Assumptions for security of ElGamal

## Decisional Diffie–Hellman assumption (DDH)

The following two probability distributions are computationally indistinguishable (with a PPTA in $\log q$):

- $(g^a, g^b, g^{ab})$ where $a$ and $b$ are randomly and independently chosen.
- $(g^a, g^b, g^c)$ where $a$, $b$ and $c$ are randomly and independently chosen.

DDH is considered stronger than CDH:
If CDH is false then one can compute with a PPTA $g^{ab}$ from $g^a, g^b$, and so DDH is false.

# DDH false on multiplicative group

## Euler's criterion

Given a prime $p$ and $a$ an integer coprime to $p$

$$a^{\frac{p-1}{2}} \equiv \begin{cases} 1 \pmod{p} & \text{if there is } x \text{ s.t. } a \equiv x^2 \pmod{p}, \\ -1 \pmod{p} & \text{if there is no such integer.} \end{cases}$$

$\Rightarrow$ easy to compute for $g^y$ and so determine if $y$ is odd or even.

Given $g^a$, $g^b$ and $g^{ab}$, one can compare the least significant bit of $a$, $b$ and $ab$, and distinguish $g^{ab}$ from a random group element.

# Group used by Belenios

Group for which Discrete log, CDH, DDH are assumed true.

## Schnorr group

- subgroup of the multiplicative group of integers modulo $p$
- $p$, $q$ primes and $r$ integer s.t. $p = qr + 1$
- $g = h^r \pmod{p}$ with $h$ s.t. $h^r \not\equiv 1 \pmod{p}$

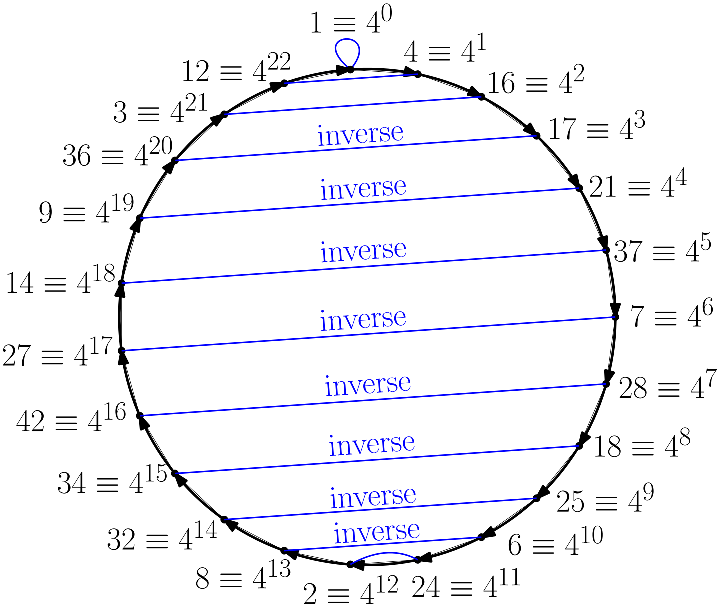
## BELENIOS-2048

p = 20694785691422546
401013643657505008064922989295751104097100884787057374219242
717401922237254497684338129066633138078958404960054389636289
796393038773905722803605973749427671376777618898589872735865
049081167099310535867780980030790491654063777173764198678527
273474476341835600035698305193144284561701911000786737307333
564123971732897913240474578834468260652327974647951137672658
693582180046317922073668860052627186363386088796882120769432
366149491002923444346373222145884100586421050242120365433561
201320481118852408731077014151666200162313177169372189248078
507711827842317498073276598828825169183103125680162072880719

g = 2402352677501852
209227687703532399932712287657378364916510075318787663274146
353219320285676155269678799694668298749389095083896573425601
900601068477164491735474137283104610458681314511781646755400
527402889846139864532661215055797097162016168270312886432456
663834863635782106154918419982534315189740658186868651151358
576410138882215396016043228843603930989333662772848406593138
406010231675095763777982665103606822406635076697764025346253
773085133173495194248967754052573659049492477631475991575198
775177711481490920456600205478127054728238140972518639858334
115700568353695553423781475582491896050296680037745308460627

q = 78571733251071885
079927659812671450121821421258408794611510081919805623223441

# Group $\pmod{47}$ with $g = 4$ and $q = 23$



# Partial conclusion

# Partial conclusion

We have seen the ELGamal encryption scheme used by Belenios

Next week, we will see:

- more cryptographic tools: Cryptographic hash function, Zero-Knowledge Proofs, Schnorr signature scheme, Pedersen's threshold secret sharing scheme
- more details on how Belenios works
- more details on how it is implemented