Bern University
of Applied Sciences

An Alternative Group for Applications of ElGamal in Cryptographic Protocols

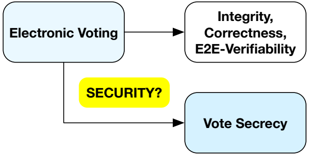
Rolf Haenni & Ilona Starý Kořánová

E-Vote-ID 2023, Luxembourg, October 3, 2023

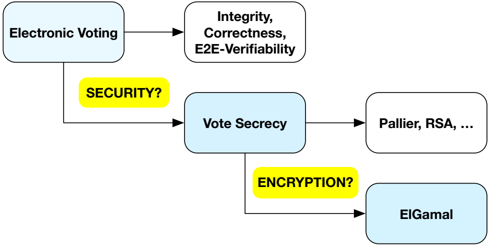
Introduction

Electronic Voting

Introduction



Introduction



Electronic Voting

Integrity,
Correctness,
E2E-Verifiability

SECURITY?

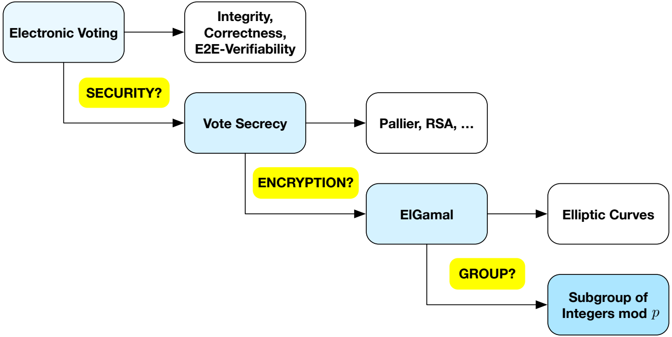
Vote Secrecy

Pallier, RSA, ...

ENCRYPTION?

ElGamal

Introduction



Electronic Voting

Integrity, Correctness, E2E-Verifiability

SECURITY?

Vote Secrecy

Pallier, RSA, ...

ENCRYPTION?

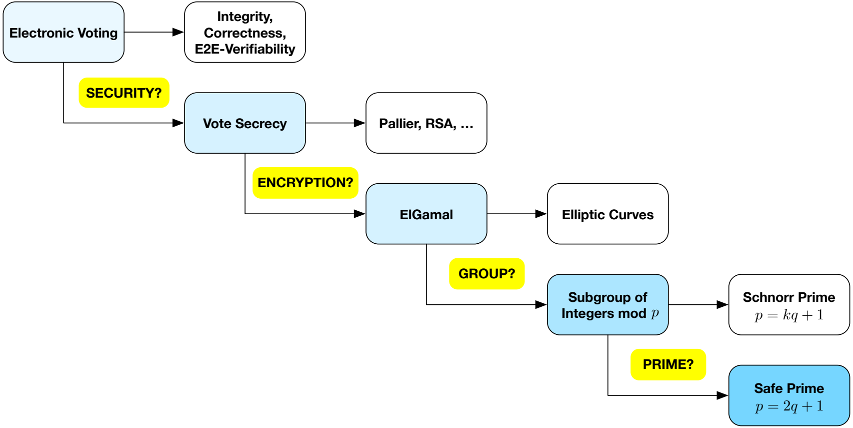
ElGamal

Elliptic Curves

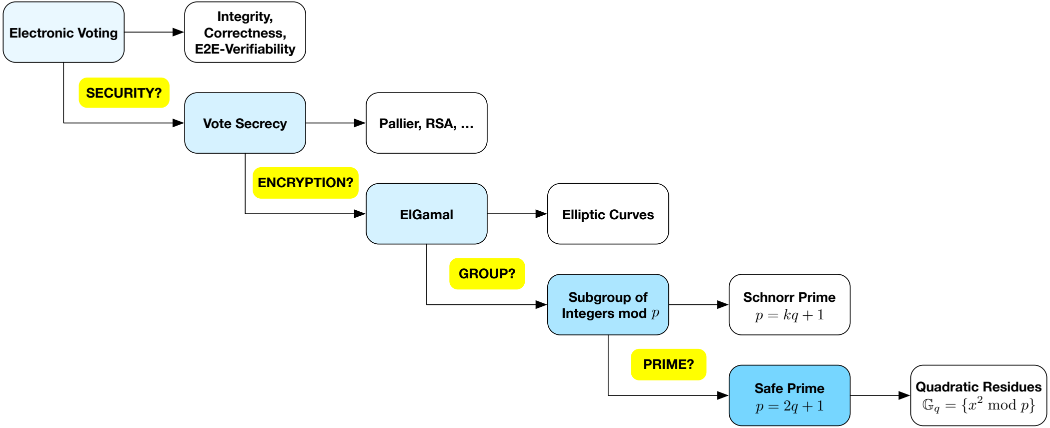
GROUP?

Subgroup of Integers mod p

Introduction



Introduction



Electronic Voting

Integrity, Correctness, E2E-Verifiability

SECURITY?

Vote Secrecy

Pallier, RSA, ...

ENCRYPTION?

ElGamal

Elliptic Curves

GROUP?

Subgroup of Integers mod p

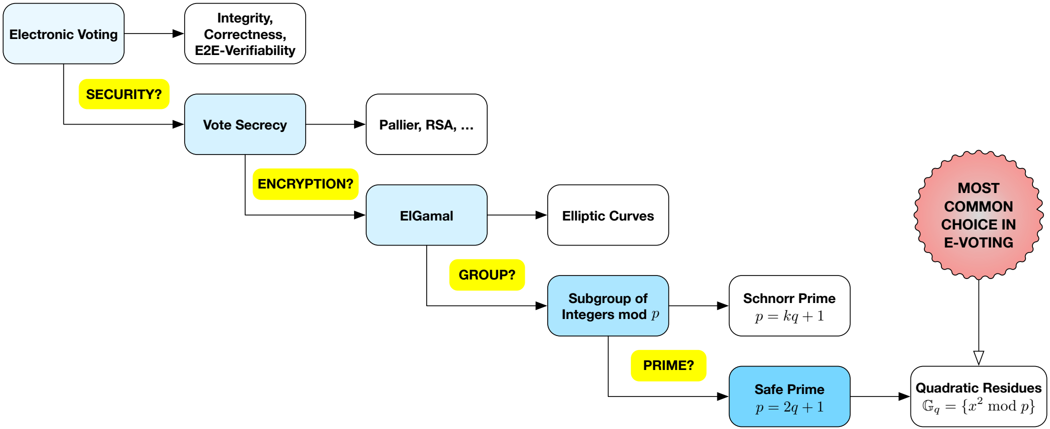
Schnorr Prime $p = kq + 1$

PRIME?

Safe Prime $p = 2q + 1$

Quadratic Residues $\mathbb{G}_q = \{x^2 \bmod p\}$

Introduction



Electronic Voting

Integrity, Correctness, E2E-Verifiability

SECURITY?

Vote Secrecy

Pallier, RSA, ...

ENCRYPTION?

ElGamal

Elliptic Curves

GROUP?

Subgroup of Integers mod p

Schnorr Prime $p = kq + 1$

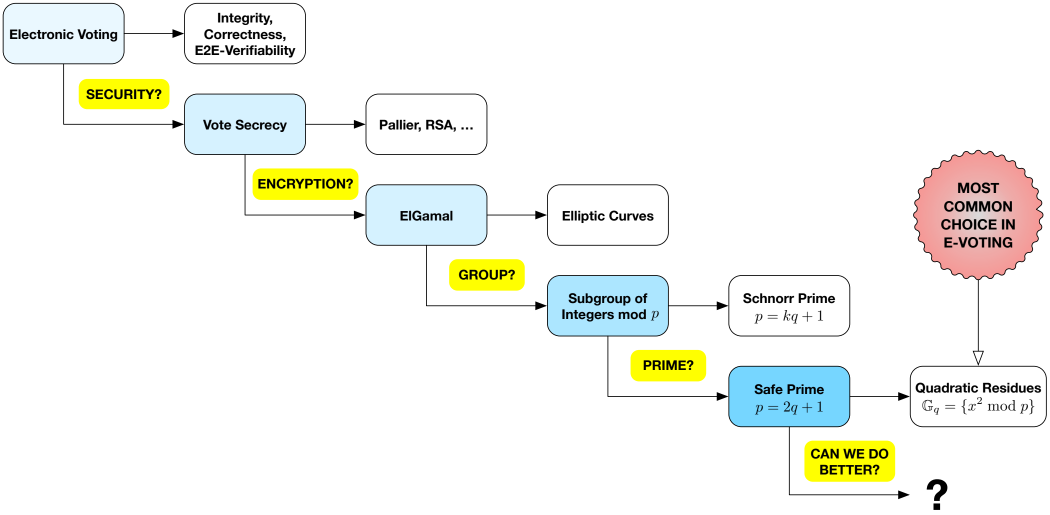
PRIME?

Safe Prime $p = 2q + 1$

Quadratic Residues $\mathbb{G}_q = \{x^2 \bmod p\}$

MOST COMMON CHOICE IN E-VOTING

Introduction



Electronic Voting

Integrity, Correctness, E2E-Verifiability

SECURITY?

Vote Secrecy

Pallier, RSA, ...

ENCRYPTION?

ElGamal

Elliptic Curves

GROUP?

Subgroup of Integers mod p

Schnorr Prime $p = kq + 1$

PRIME?

Safe Prime $p = 2q + 1$

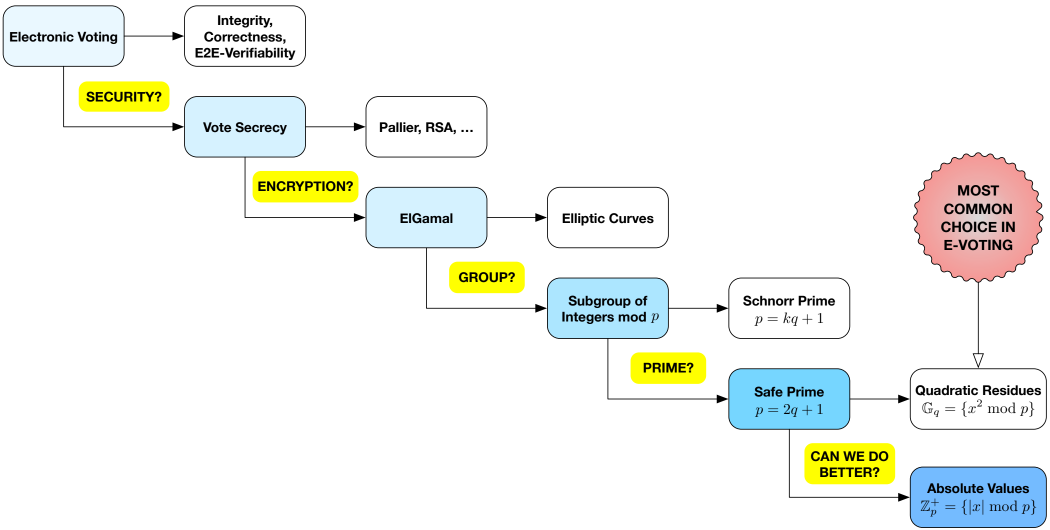
Quadratic Residues $\mathbb{G}_q = \{x^2 \bmod p\}$

MOST COMMON CHOICE IN E-VOTING

CAN WE DO BETTER?

?

Introduction

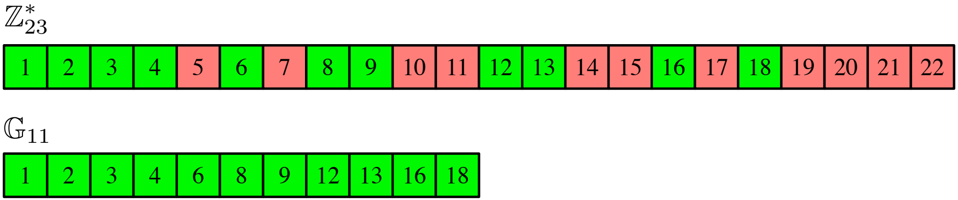

Electronic Voting
Integrity, Correctness, E2E-Verifiability
SECURITY?
Vote Secrecy
Pallier, RSA, ...
ENCRYPTION?
ElGamal
Elliptic Curves
MOST COMMON CHOICE IN E-VOTING
GROUP?
Subgroup of Integers mod p
Schnorr Prime $p = kq + 1$
PRIME?
Safe Prime $p = 2q + 1$
Quadratic Residues $\mathbb{G}_q = \{x^2 \bmod p\}$
CAN WE DO BETTER?
Absolute Values $\mathbb{Z}_p^+ = \{|x| \bmod p\}$

Subgroup of Quadratic Residues

- Let $p = 2q + 1$ be a safe prime and $\mathbb{Z}_p^* = \{1, \ldots, p-1\}$
- Let $\mathbb{G}_q = \{x^2 \bmod p : x \in \mathbb{Z}_p^*\} \subset \mathbb{Z}_p^*$ be the set of quadratic residues

Subgroup of Quadratic Residues

- Let $p = 2q + 1$ be a safe prime and $\mathbb{Z}_p^* = \{1, \ldots, p-1\}$
- Let $\mathbb{G}_q = \{x^2 \bmod p : x \in \mathbb{Z}_p^*\} \subset \mathbb{Z}_p^*$ be the set of quadratic residues
- Example: $p = 2 * 11 + 1 = 23$ and $q = 11$



$\mathbb{Z}_{23}^*$

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22

$\mathbb{G}_{11}$

1	2	3	4	6	8	9	12	13	16	18

Properties of $\mathbb{G}_q$

- Subgroup membership
  - Square numbers $1, 4, 9, 16, 25, \ldots$ are always members of $\mathbb{G}_q$
  - 3 is always a member of $\mathbb{G}_q$
  - But: membership of $2, 5, 6, 7, \ldots$ depends on p
  - $x \in \mathbb{G}_q$ implies $p - x \notin \mathbb{G}_q$ (and vice versa)

Properties of $\mathbb{G}_q$

- Subgroup membership
  - Square numbers $1, 4, 9, 16, 25, \ldots$ are always members of $\mathbb{G}_q$
  - 3 is always a member of $\mathbb{G}_q$
  - But: membership of $2, 5, 6, 7, \ldots$ depends on p
  - $x \in \mathbb{G}_q$ implies $p - x \notin \mathbb{G}_q$ (and vice versa)
- All elements of $\mathbb{G}_q \setminus \{1\}$ are generators

Properties of $\mathbb{G}_q$

- Subgroup membership
  - Square numbers $1, 4, 9, 16, 25, \ldots$ are always members of $\mathbb{G}_q$
  - 3 is always a member of $\mathbb{G}_q$
  - But: membership of $2, 5, 6, 7, \ldots$ depends on p
  - $x \in \mathbb{G}_q$ implies $p - x \notin \mathbb{G}_q$ (and vice versa)
- All elements of $\mathbb{G}_q \setminus \{1\}$ are generators
- Testing subgroup membership $x \in \mathbb{G}_q$
  - Method 1: check if $x^q \bmod p = 1$
  - Method 2: check if $\left(\frac{x}{p}\right) = 1$

Practical Disadvantages of $\mathbb{G}_q$

- Testing group membership is relatively expensive
  - $1\times$modular exponentiation for Method 1
  - $1\times$Jacobi symbol for Method 2

Practical Disadvantages of $\mathbb{G}_q$

- Testing group membership is relatively expensive
  - $1\times$modular exponentiation for Method 1
  - $1\times$Jacobi symbol for Method 2
- Group membership depends on p
  - Selecting generators
  - Generating random group elements

Practical Disadvantages of $\mathbb{G}_q$

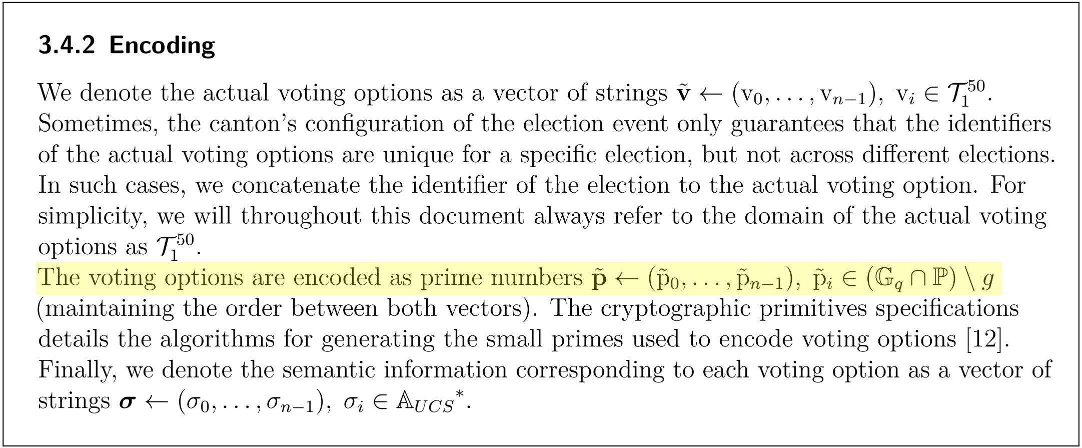
- Testing group membership is relatively expensive
  - $1\times$modular exponentiation for Method 1
  - $1\times$Jacobi symbol for Method 2
- Group membership depends on p
  - Selecting generators
  - Generating random group elements
- ElGamal message encoding $\Gamma : \mathcal{M} \rightarrow \mathbb{G}_q$ depends on p
  - General-purpose messages $\mathcal{M} = \{0,1\}^n$
  - Specific messages $\mathcal{M} = \{m_1, \ldots, m_n\}$
  - Example: prime number encoding of voting options (Norway, Swiss Post, CHVote, . . .)

How Expensive is Group Membership Testing?

	$\|\|p\|\| = 2048$ bits		$\|\|p\|\| = 3072$ bits	
	$\left(\frac{x}{p}\right) = 1$	$x^q \bmod p = 1$	$\left(\frac{x}{p}\right) = 1$	$x^q \bmod p = 1$
C (GMP)	98ms	23’224ms	186ms	72’992ms
Java (Bouncy Castle)	12’871ms	35’705ms	27’132ms	114’262ms
Python (SymPi)	15’447ms	243’561ms	34’762ms	691’568ms
Javascript (VJSC)	12’453ms	692’821ms	23’878ms	2’162’474ms

Running times measured for 10’000 membership tests (MacBook Pro, 2.3GHz i9)

How Difficult is Message Encoding in Practice?

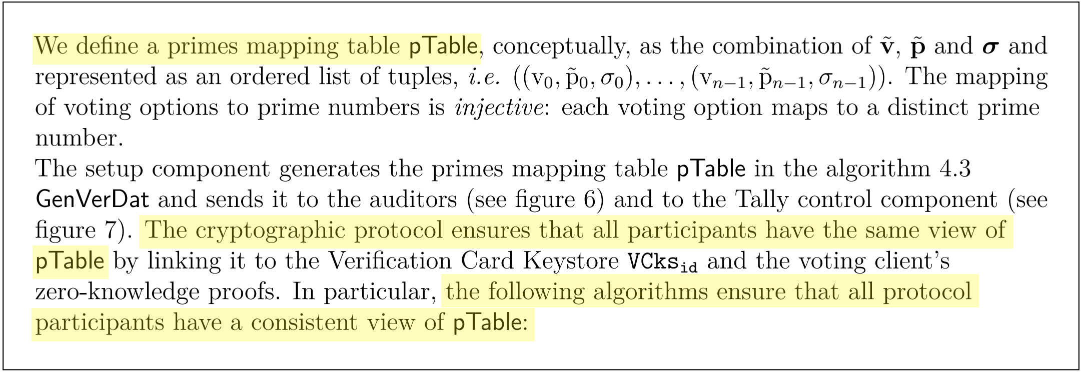


3.4.2 Encoding

We denote the actual voting options as a vector of strings $\tilde{\mathbf{v}} \leftarrow (\mathrm{v}_0, \ldots, \mathrm{v}_{n-1}),\ \mathrm{v}_i \in \mathcal{T}_1^{50}$. Sometimes, the canton's configuration of the election event only guarantees that the identifiers of the actual voting options are unique for a specific election, but not across different elections. In such cases, we concatenate the identifier of the election to the actual voting option. For simplicity, we will throughout this document always refer to the domain of the actual voting options as $\mathcal{T}_1^{50}$.
The voting options are encoded as prime numbers $\tilde{\mathbf{p}} \leftarrow (\tilde{\mathrm{p}}_0, \ldots, \tilde{\mathrm{p}}_{n-1}),\ \tilde{\mathrm{p}}_i \in (\mathbb{G}_q \cap \mathbb{P}) \setminus g$ (maintaining the order between both vectors). The cryptographic primitives specifications details the algorithms for generating the small primes used to encode voting options [12]. Finally, we denote the semantic information corresponding to each voting option as a vector of strings $\boldsymbol{\sigma} \leftarrow (\sigma_0, \ldots, \sigma_{n-1}),\ \sigma_i \in \mathbb{A}_{UCS}{}^*$.

"Swiss Post Voting System – System Specification, Version 1.3.1" (Page 21–22, Part I)

How Difficult is Message Encoding in Practice?



We define a primes mapping table pTable, conceptually, as the combination of $\tilde{\mathbf{v}}$, $\tilde{\mathbf{p}}$ and $\boldsymbol{\sigma}$ and represented as an ordered list of tuples, *i.e.* $((\mathrm{v}_0, \tilde{\mathrm{p}}_0, \sigma_0), \ldots, (\mathrm{v}_{n-1}, \tilde{\mathrm{p}}_{n-1}, \sigma_{n-1}))$. The mapping of voting options to prime numbers is *injective*: each voting option maps to a distinct prime number.

The setup component generates the primes mapping table pTable in the algorithm 4.3 GenVerDat and sends it to the auditors (see figure 6) and to the Tally control component (see figure 7). The cryptographic protocol ensures that all participants have the same view of pTable by linking it to the Verification Card Keystore $\mathtt{VCks}_{\mathtt{id}}$ and the voting client's zero-knowledge proofs. In particular, the following algorithms ensure that all protocol participants have a consistent view of pTable:

"Swiss Post Voting System – System Specification, Version 1.3.1" (Page 21–22, Part II)

How Difficult is Message Encoding in Practice?

- The setup component includes the contents of pTable in the authenticated symmetric encryption in 4.9 GenCredDat.
- The voting client includes the contents of pTable in the authenticated symmetric decryption in 5.3 GetKey and in the zero-knowledge proofs in 5.4 CreateVote.
- The control components include the contents of pTable when verifying the voting client's zero-knowledge proofs in 5.5 VerifyBallotCCR$_j$.
- The tally control component includes the contents of pTable when verifying the voting client's zero-knowledge proofs in 6.4 VerifyVotingClientProofs.
- The auditors include the contents of pTable when verifying the voting client's zero-knowledge proofs in VerifyTally.

"Swiss Post Voting System – System Specification, Version 1.3.1" (Page 21–22, Part III)

Proposal For a Alternative Group

- Definition: let $|x| \overset{\text{def.}}{=} \min(x, p - x)$ be the absolute value of $x \in \mathbb{Z}_p^*$
- Let $\mathbb{Z}_p^+ = \{|x| : x \in \mathbb{Z}_p^*\} = \{1, \ldots, q\}$

Proposal For a Alternative Group

- Definition: let $|x| \overset{\text{def.}}{=} \min(x, p-x)$ be the absolute value of $x \in \mathbb{Z}_p^*$
- Let $\mathbb{Z}_p^+ = \{|x| : x \in \mathbb{Z}_p^*\} = \{1, \ldots, q\}$
- Let $x \circ y = |xy \bmod p|$ and $\mathrm{inv}(x) = |x^{-1} \bmod p|$

Proposal For a Alternative Group

- Definition: let $|x| \overset{\text{def.}}{=} \min(x, p-x)$ be the absolute value of $x \in \mathbb{Z}_p^*$
- Let $\mathbb{Z}_p^+ = \{|x| : x \in \mathbb{Z}_p^*\} = \{1, \ldots, q\}$
- Let $x \circ y = |xy \bmod p|$ and $\mathrm{inv}(x) = |x^{-1} \bmod p|$
- $(\mathbb{Z}_p^+, \circ, \mathrm{inv}, 1)$ forms a group: the group of absolute values modulo $p = 2q + 1$
- Proof: see paper

Alternative Definition (for Mathematicians)

- Consider the trivial subgroup $\mathbb{G}_2 = \{1, p-1\}$

Alternative Definition (for Mathematicians)

- Consider the trivial subgroup $\mathbb{G}_2 = \{1, p-1\}$
- Since modular multiplication is commutative, it follows that $\mathbb{G}_2$ is normal

Alternative Definition (for Mathematicians)

- Consider the trivial subgroup $\mathbb{G}_2 = \{1, p-1\}$
- Since modular multiplication is commutative, it follows that $\mathbb{G}_2$ is normal
- Therefore, $\mathbb{Z}_p^*/\mathbb{G}_2 = \{a\mathbb{G}_2 : a \in \mathbb{Z}_p^*\} = \{\{1, p-1\}, \{2, p-2\}, \ldots, \{q, q+1\}\}$ forms a quotient group of order q

Alternative Definition (for Mathematicians)

- Consider the trivial subgroup $\mathbb{G}_2 = \{1, p-1\}$
- Since modular multiplication is commutative, it follows that $\mathbb{G}_2$ is normal
- Therefore, $\mathbb{Z}_p^*/\mathbb{G}_2 = \{a\mathbb{G}_2 : a \in \mathbb{Z}_p^*\} = \{\{1, p-1\}, \{2, p-2\}, \ldots, \{q, q+1\}\}$ forms a quotient group of order q
- Let every coset $\{x, p-x\} \in \mathbb{Z}_p^*/\mathbb{G}_2$ be represented by $x = \min(x, p-x)$

Alternative Definition (for Mathematicians)

- Consider the trivial subgroup $\mathbb{G}_2 = \{1, p-1\}$
- Since modular multiplication is commutative, it follows that $\mathbb{G}_2$ is normal
- Therefore, $\mathbb{Z}_p^*/\mathbb{G}_2 = \{a\mathbb{G}_2 : a \in \mathbb{Z}_p^*\} = \{\{1, p-1\}, \{2, p-2\}, \ldots, \{q, q+1\}\}$ forms a quotient group of order q
- Let every coset $\{x, p-x\} \in \mathbb{Z}_p^*/\mathbb{G}_2$ be represented by $x = \min(x, p-x)$
- This implies $\mathbb{Z}_p^*/\mathbb{G}_2 \equiv \{1, \ldots, q\}$ and therefore $\mathbb{Z}_p^*/\mathbb{G}_2 \equiv \mathbb{Z}_p^+$

Alternative Definition (for Mathematicians)

- Consider the trivial subgroup $\mathbb{G}_2 = \{1, p-1\}$
- Since modular multiplication is commutative, it follows that $\mathbb{G}_2$ is normal
- Therefore, $\mathbb{Z}_p^*/\mathbb{G}_2 = \{a\mathbb{G}_2 : a \in \mathbb{Z}_p^*\} = \{\{1, p-1\}, \{2, p-2\}, \ldots, \{q, q+1\}\}$ forms a quotient group of order q
- Let every coset $\{x, p-x\} \in \mathbb{Z}_p^*/\mathbb{G}_2$ be represented by $x = \min(x, p-x)$
- This implies $\mathbb{Z}_p^*/\mathbb{G}_2 \equiv \{1, \ldots, q\}$ and therefore $\mathbb{Z}_p^*/\mathbb{G}_2 \equiv \mathbb{Z}_p^+$

Properties of $\mathbb{Z}_p^+$

- Exponentiations in $\mathbb{Z}_p^+$ can be be computed efficiently as $|x^y \bmod p \bmod p|$

Properties of $\mathbb{Z}_p^+$

- Exponentiations in $\mathbb{Z}_p^+$ can be be computed efficiently as $|x^y \bmod p \bmod p|$
- From $|\mathbb{Z}_p^+| = |\mathbb{G}_q| = q$, it follows that $\mathbb{Z}_p^+$ is isomorphic to $\mathbb{G}_q$

Properties of $\mathbb{Z}_p^+$

- Exponentiations in $\mathbb{Z}_p^+$ can be be computed efficiently as $|x^y \bmod p \bmod p|$
- From $|\mathbb{Z}_p^+| = |\mathbb{G}_q| = q$, it follows that $\mathbb{Z}_p^+$ is isomorphic to $\mathbb{G}_q$
- The isomorphism $\phi : \mathbb{Z}_p^+ \rightarrow \mathbb{G}_q$ can be computed efficiently in both directions
  - $\phi(x) = x^2 \bmod p$, for $x \in \mathbb{Z}_p^+$
  - $\phi^{-1}(y) = |\sqrt{y} \bmod p \bmod p| = |y^{\frac{q+1}{2}} \bmod p \bmod p|$, for $y \in \mathbb{G}_q$

Properties of $\mathbb{Z}_p^+$

- Exponentiations in $\mathbb{Z}_p^+$ can be be computed efficiently as $|x^y \bmod p \bmod p|$
- From $|\mathbb{Z}_p^+| = |\mathbb{G}_q| = q$, it follows that $\mathbb{Z}_p^+$ is isomorphic to $\mathbb{G}_q$
- The isomorphism $\phi : \mathbb{Z}_p^+ \rightarrow \mathbb{G}_q$ can be computed efficiently in both directions
    - $\phi(x) = x^2 \bmod p$, for $x \in \mathbb{Z}_p^+$
    - $\phi^{-1}(y) = |\sqrt{y} \bmod p \bmod p| = |y^{\frac{q+1}{2}} \bmod p \bmod p|$, for $y \in \mathbb{G}_q$
- Therefore, if DDH is hard in $\mathbb{G}_q$, it is equally hard in $\mathbb{Z}_p^+$

Properties of $\mathbb{Z}_p^+$

- Exponentiations in $\mathbb{Z}_p^+$ can be be computed efficiently as $|x^y \bmod p \bmod p|$
- From $|\mathbb{Z}_p^+| = |\mathbb{G}_q| = q$, it follows that $\mathbb{Z}_p^+$ is isomorphic to $\mathbb{G}_q$
- The isomorphism $\phi : \mathbb{Z}_p^+ \rightarrow \mathbb{G}_q$ can be computed efficiently in both directions
  - $\phi(x) = x^2 \bmod p$, for $x \in \mathbb{Z}_p^+$
  - $\phi^{-1}(y) = |\sqrt{y} \bmod p \bmod p| = |y^{\frac{q+1}{2}} \bmod p \bmod p|$, for $y \in \mathbb{G}_q$
- Therefore, if DDH is hard in $\mathbb{G}_q$, it is equally hard in $\mathbb{Z}_p^+$

Practical Advantages of $\mathbb{Z}_p^+$ Over $\mathbb{G}_q$

- Group membership $x \in \mathbb{Z}_p^+$ can be tested efficiently as $1 \leq x \leq q$

Practical Advantages of $\mathbb{Z}_p^+$ Over $\mathbb{G}_q$

- Group membership $x \in \mathbb{Z}_p^+$ can be tested efficiently as $1 \leq x \leq q$
- Since $p < p'$ implies $\mathbb{Z}_p^+ \subset \mathbb{Z}_{p'}^+$, it follows that:
  - $1, 2, 3, 4, 5, \ldots$ are always group elements,
  - $1, 2, 3, 4, 5, \ldots$ are possible random group elements,
  - $2, 3, 4, 5, 6, \ldots$ are always generators,
  - $2, 3, 5, 7, 11, \ldots$ are always the smallest prime elements,

  i.e., independently of p

Practical Advantages of $\mathbb{Z}_p^+$ Over $\mathbb{G}_q$

- Group membership $x \in \mathbb{Z}_p^+$ can be tested efficiently as $1 \leq x \leq q$
- Since $p < p'$ implies $\mathbb{Z}_p^+ \subset \mathbb{Z}_{p'}^+$, it follows that:
  - $1, 2, 3, 4, 5, \ldots$ are always group elements,
  - $1, 2, 3, 4, 5, \ldots$ are possible random group elements,
  - $2, 3, 4, 5, 6, \ldots$ are always generators,
  - $2, 3, 5, 7, 11, \ldots$ are always the smallest prime elements,

  i.e., independently of p
- For general-purpose messages, $\Gamma : \{0,1\}^n \rightarrow \mathbb{Z}_p^+$ can be defined by interpreting them as binary numbers (except for 0^n)

Practical Advantages of $\mathbb{Z}_p^+$ Over $\mathbb{G}_q$

- Group membership $x \in \mathbb{Z}_p^+$ can be tested efficiently as $1 \leq x \leq q$
- Since $p < p'$ implies $\mathbb{Z}_p^+ \subset \mathbb{Z}_{p'}^+$, it follows that:
  - $1, 2, 3, 4, 5, \ldots$ are always group elements,
  - $1, 2, 3, 4, 5, \ldots$ are possible random group elements,
  - $2, 3, 4, 5, 6, \ldots$ are always generators,
  - $2, 3, 5, 7, 11, \ldots$ are always the smallest prime elements,

  i.e., independently of p
- For general-purpose messages, $\Gamma : \{0,1\}^n \rightarrow \mathbb{Z}_p^+$ can be defined by interpreting them as binary numbers (except for 0^n)
- For specific messages, $\Gamma : \{m_1, \ldots, m_n\} \rightarrow \mathbb{Z}_p^+$ can be defined “globally”, i.e., independently of p

Conclusion

- From a security perspective, $\mathbb{Z}_p^+$ and $\mathbb{G}_q$ are equivalent (DDH holds)
- Group operation in $\mathbb{Z}_p^+$ is slightly less efficient (but cost is negligible)
- Membership tests in $\mathbb{Z}_p^+$ are much more efficient
- Plus some other practical advantages

Conclusion

- From a security perspective, $\mathbb{Z}_p^+$ and $\mathbb{G}_q$ are equivalent (DDH holds)
- Group operation in $\mathbb{Z}_p^+$ is slightly less efficient (but cost is negligible)
- Membership tests in $\mathbb{Z}_p^+$ are much more efficient
- Plus some other practical advantages
- General recommendation:

Use $\mathbb{Z}_p^+$ instead of $\mathbb{G}_q$ in applications and implementations of ElGamal

PS: Already implemented in CHVote 1.3