An Homomorphic LWE based E-voting Scheme

I. Chillotti[1] N. Gama[1,2] M. Georgieva[3] M. Izabachène[4]

1 LMV Laboratoire de mathématiques de Versailles - CNRS UMR 8100, Université de Versailles Saint-Quentin-en-Yvelines, université Paris-Saclay
2 inpher.io
3 gemalto
4 cea

PQCrypto 2016 - Fukuoka (Japan)
February 26, 2016

Table of contents

Table of contents

E-voting

Electronic Voting (E-voting): "would like to be" the electronic analogue of the paper voting procedure.

Common properties:

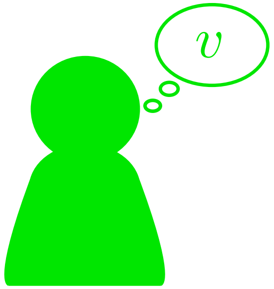
- Privacy
- Verifiability
- Correctness

Examples in the litterature:

- Mix-net based
- Homomorphic based

Post Quantum scheme

The players



Users/Voters

1. Can vote (after registration)
2. Can verify that their vote has been cast and counted

Authority A_1

1. Registration of the voters

Bulletin Board BB

1. Checks and adds the ballots
2. Performs public operations

Trustees T

1. Set up the decryption keys
2. Compute the final election result

The structure of an E-voting scheme

Setup Phase	1) Set the parameters 2) Registration of voters
Voting Phase	1) Product and send ballots 2) Process the BB
Tallying Phase	1) Decrypt the result

...and everyone can verify the result!

Table of contents

Helios: based on ElGamal

Helios relies on an additive homomorphic encryption scheme

El Gamal over a group $\langle g \rangle$ and secret s

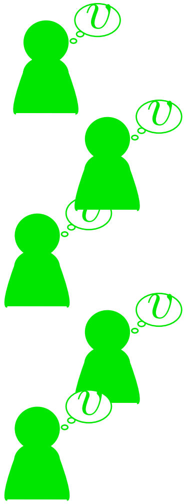
$$E(\mu) := (g^r, g^\mu g^{s \cdot r}) \text{ for some large random } r$$

Then

$$E(\mu_1) \cdot E(\mu_2) = E(\mu_1 + \mu_2)$$

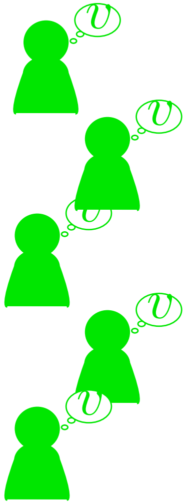
(Decryption is feasible if μ_1, μ_2 are not too large)

Main idea



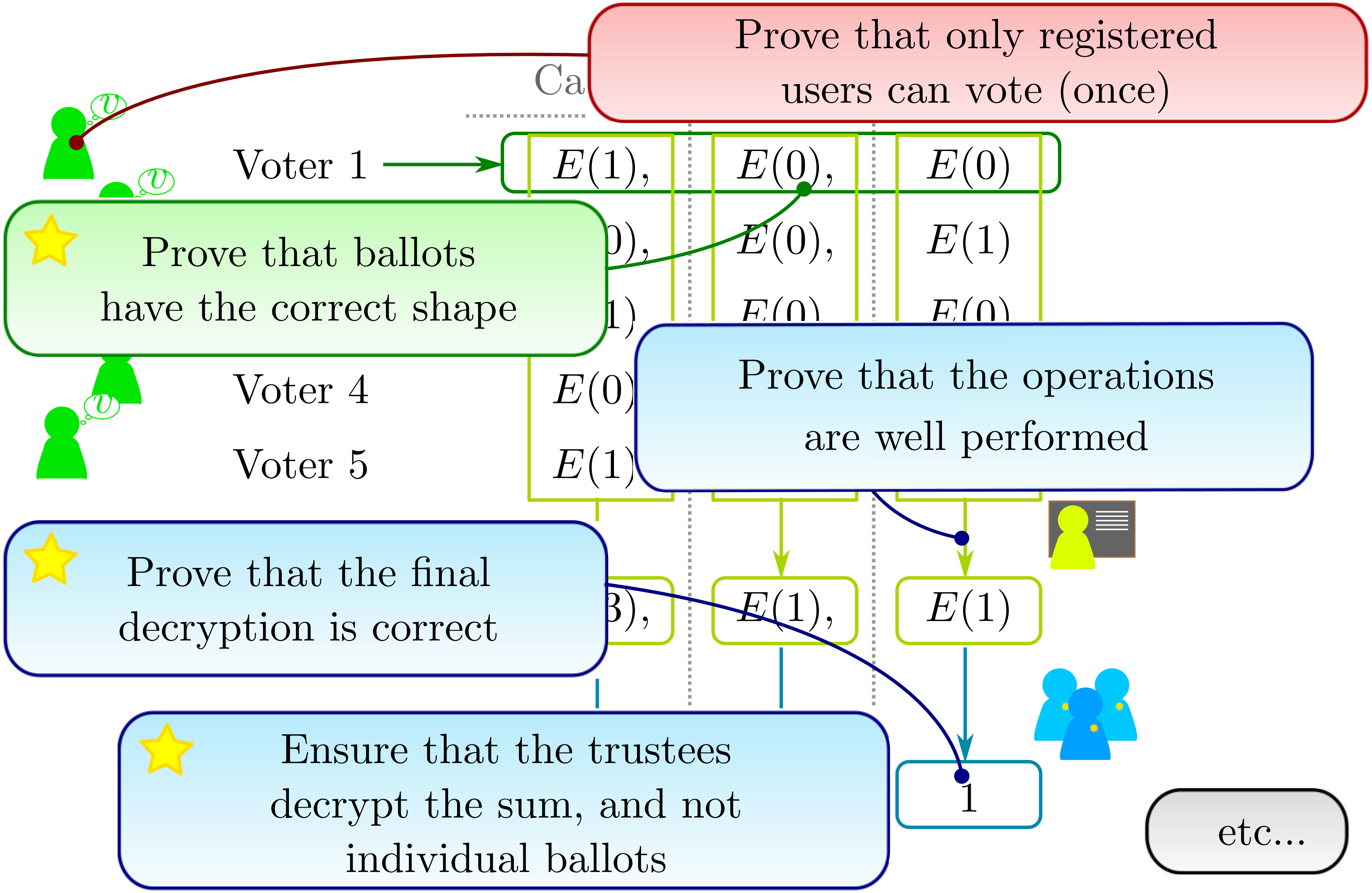
	Cand 1	Cand 2	Cand 3
Voter 1 →	$E(1)$,	$E(0)$,	$E(0)$
Voter 2 →	$E(0)$,	$E(0)$,	$E(1)$
Voter 3 →	$E(1)$,	$E(0)$,	$E(0)$
Voter 4 →	$E(0)$,	$E(1)$,	$E(0)$
Voter 5 →	$E(1)$,	$E(0)$,	$E(0)$

Main idea



	Cand 1	Cand 2	Cand 3
Voter 1	$E(1)$,	$E(0)$,	$E(0)$
Voter 2	$E(0)$,	$E(0)$,	$E(1)$
Voter 3	$E(1)$,	$E(0)$,	$E(0)$
Voter 4	$E(0)$,	$E(1)$,	$E(0)$
Voter 5	$E(1)$,	$E(0)$,	$E(0)$
(Enc. Result)	$E(3)$,	$E(1)$,	$E(1)$
(Dec. Result)	3,	1,	1

Main idea


Prove that only registered
users can vote (once)
Ca
Voter 1
Voter 4
Voter 5
E(1), E(0), E(0)
E(0), E(1)
E(0) E(0)
E(0)
E(1)
Prove that ballots
have the correct shape
Prove that the operations
are well performed
Prove that the final
decryption is correct
E(1), E(1)
Ensure that the trustees
decrypt the sum, and not
individual ballots
1
etc...

How to vote?

Helios

$v \in [0, \ell - 1]$

$(0_0, \ldots, 0_{v-1}, 1_v, 0_{v+1}, \ldots, 0_{\ell-1})$

~~$(0_0, \ldots, 0_{v-1}, 100_v, 0_{v+1}, \ldots, 0_{\ell-1})$~~

~~$(1_0, \ldots, 0_{v-1}, 1_v, 1_{v+1}, \ldots, 0_{\ell-1})$~~

To ensure that the ballot has this shape, a NIZK proof is needed!

Our protocol

$v \in [0, \ell - 1]$ with $\ell = 2^k$

Decomposition of v in base 2 :

$(b_0, \ldots, b_{k-1})$

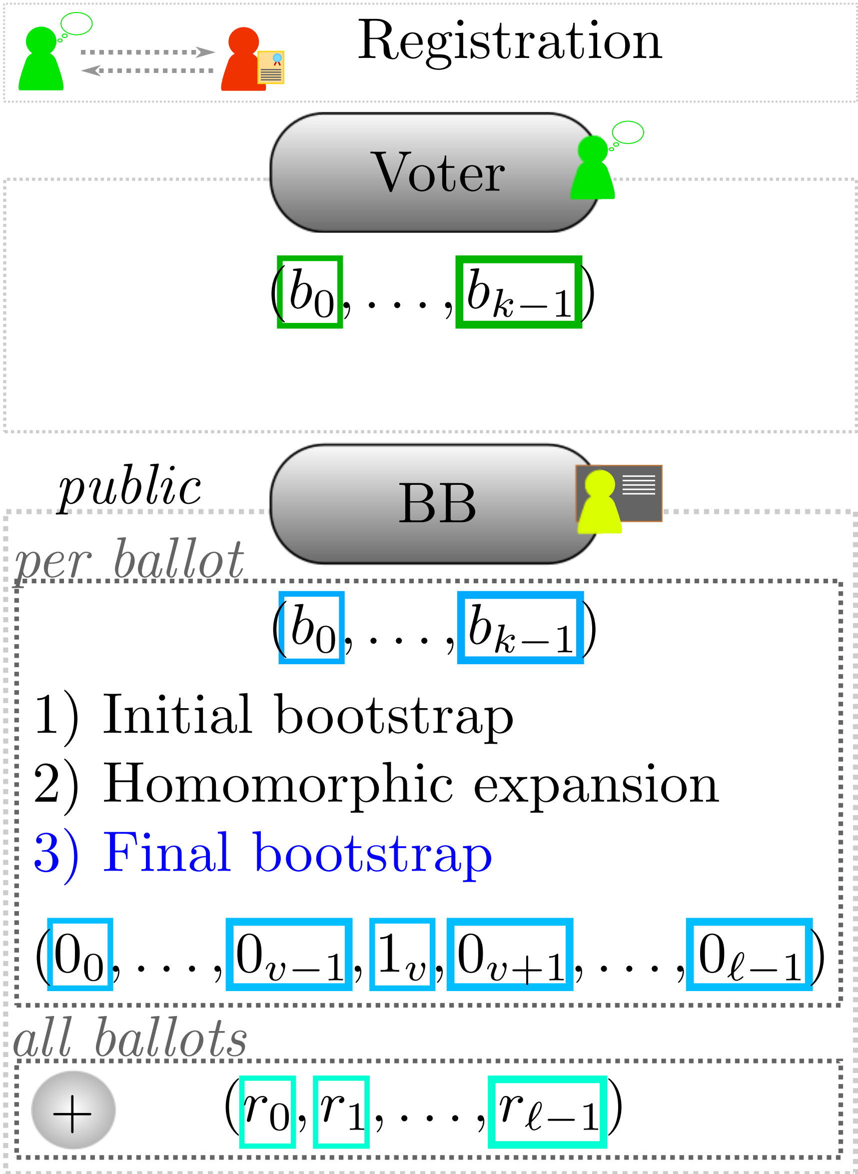
Bootstrapp

$(b_0, \ldots, b_{k-1})$

Homomorphic $(\oplus, \wedge)$

$(0_0, \ldots, 0_{v-1}, 1_v, 0_{v+1}, \ldots, 0_{\ell-1})$

Overview of the scheme



Candidate for Homomorphic Encryption

Fully Homomorphic Encryption
+
Fast Bootstrapping

Solution:

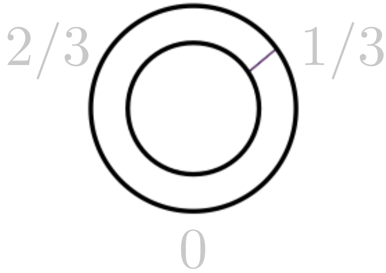
LWE-based schemes + [DM15] bootstrapping

- Final step: generalized version of [DM15]
  1. Non binary messages
  2. Lower noise amplitude

✓ Post quantum: LWE (plus other PQ blocks)

LWE

LWE Symmetric Encryption

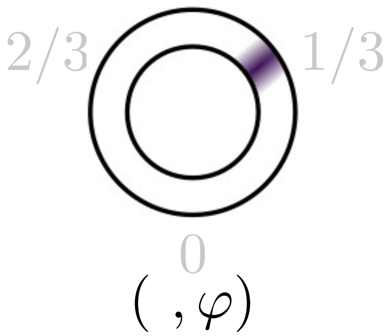


Example: $\mathcal{M} = \{0, 1/3, 2/3\}$ *mod* 1

$\mu = 1/3$ *mod* $1 \in \mathcal{M}$

LWE

LWE Symmetric Encryption



Example: $\mathcal{M} = \{0, 1/3, 2/3\}$ *mod* 1

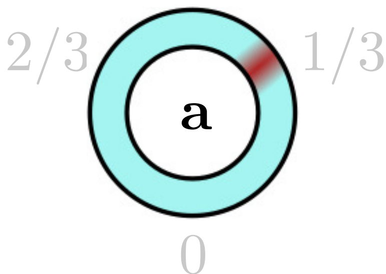
$\mu = 1/3$ *mod* $1 \in \mathcal{M}$

LWE Encryption

1. Choose $\varphi = \mu + \textit{Gaussian Error}$

LWE

LWE Symmetric Encryption



$(\mathbf{a}, \varphi)$

Example: $\mathcal{M} = \{0, 1/3, 2/3\}$ *mod* 1
$\mu = 1/3$ *mod* $1 \in \mathcal{M}$

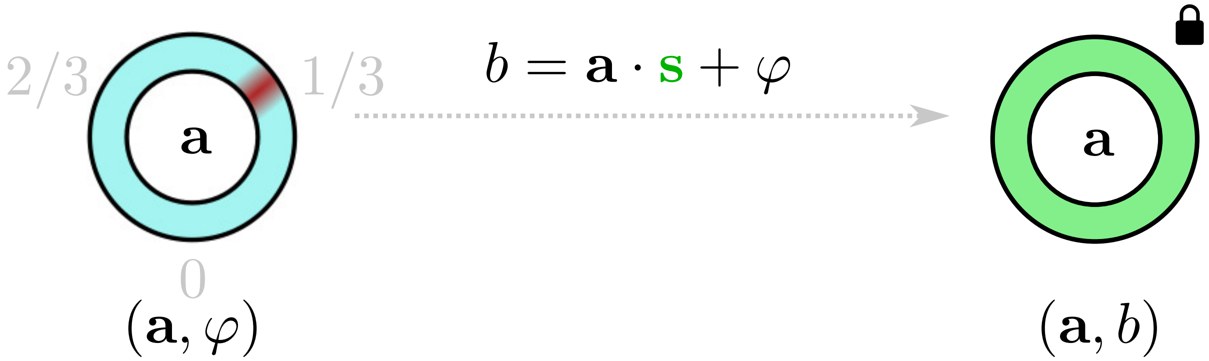
LWE Encryption

1. Choose $\varphi = \mu + \textit{Gaussian Error}$
2. Choose a random mask $\mathbf{a} \in \mathbb{T}^n$

LWE

LWE Symmetric Encryption

secret key: $\mathbf{s} \in \{0,1\}^n$



Example: $\mathcal{M} = \{0, 1/3, 2/3\}$ *mod* 1

$\mu = 1/3$ *mod* $1 \in \mathcal{M}$

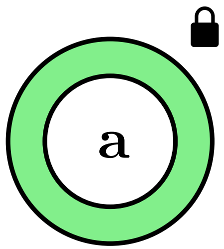
LWE Encryption

1. Choose $\varphi = \mu + \textit{Gaussian Error}$
2. Choose a random mask $\mathbf{a} \in \mathbb{T}^n$
3. Return the locked representation $(\mathbf{a}, b)$

LWE

LWE Symmetric Encryption

secret key: $\mathbf{s} \in \{0,1\}^n$



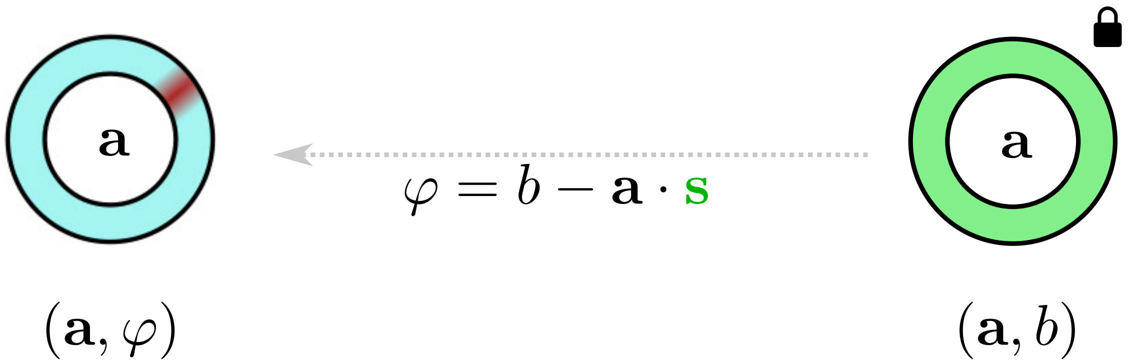
$(\mathbf{a}, b)$

LWE Decryption

LWE

LWE Symmetric Encryption

secret key: $\mathbf{s} \in \{0,1\}^n$



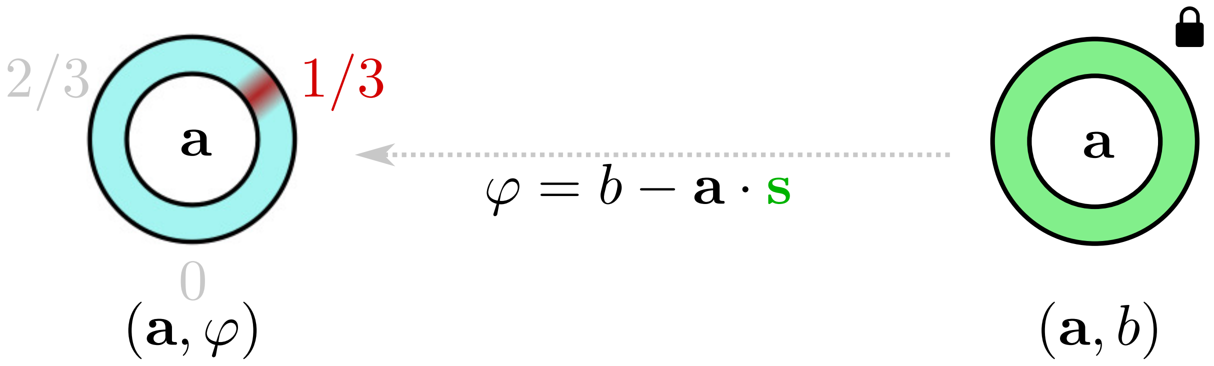
LWE Decryption

1. Unlock the representation $(\mathbf{a}, \varphi)$

LWE

LWE Symmetric Encryption

secret key: $\mathbf{s} \in \{0,1\}^n$



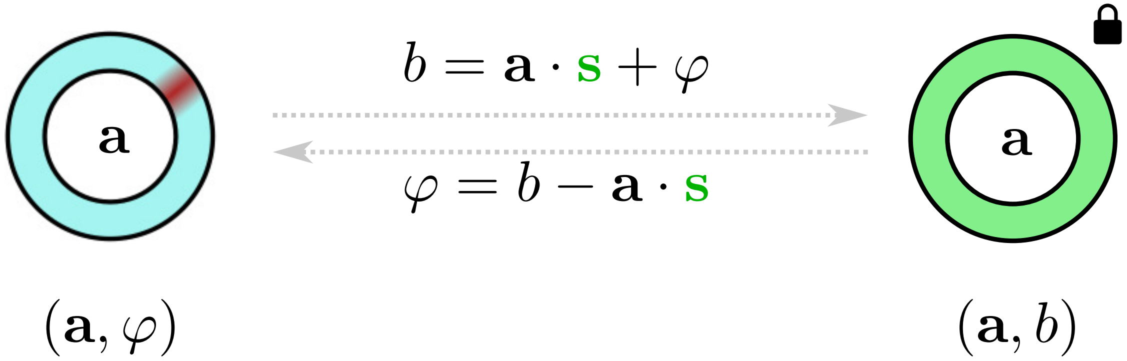
LWE Decryption

1. Unlock the representation $(\mathbf{a}, \varphi)$
2. Round φ to the nearest message $\mu \in \mathcal{M}$

LWE

LWE Symmetric Encryption

secret key: $\mathbf{s} \in \{0,1\}^n$



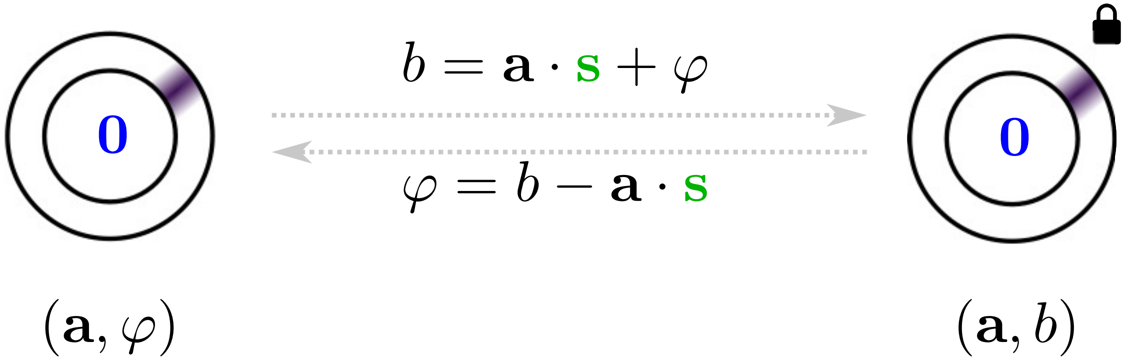
Trivial LWE samples

- LWE samples with mask $\mathbf{a} = \mathbf{0}$ are trivial.

LWE

LWE Symmetric Encryption

secret key: $\mathbf{s} \in \{0,1\}^n$



Trivial LWE samples

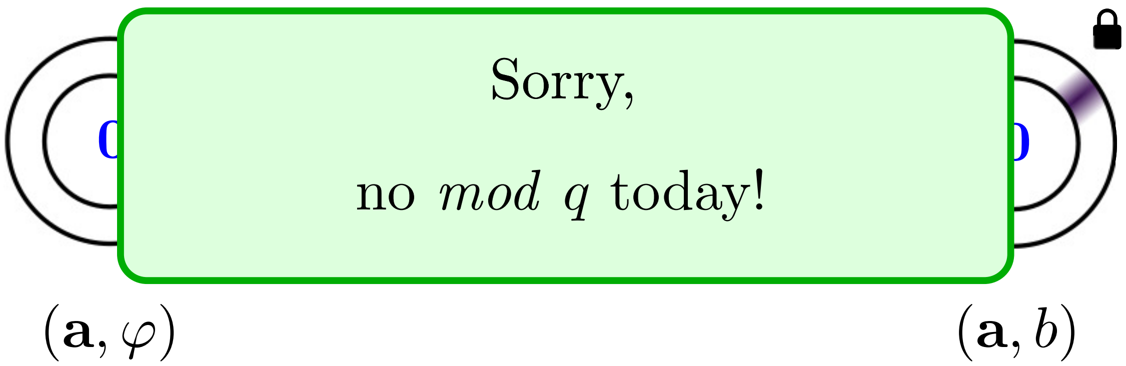
- LWE samples with mask $\mathbf{a} = \mathbf{0}$ are trivial.
- They never occur in general

...but are still worth mentionning!

LWE

LWE Symmetric Encryption

secret key: $\mathbf{s} \in \{0,1\}^n$



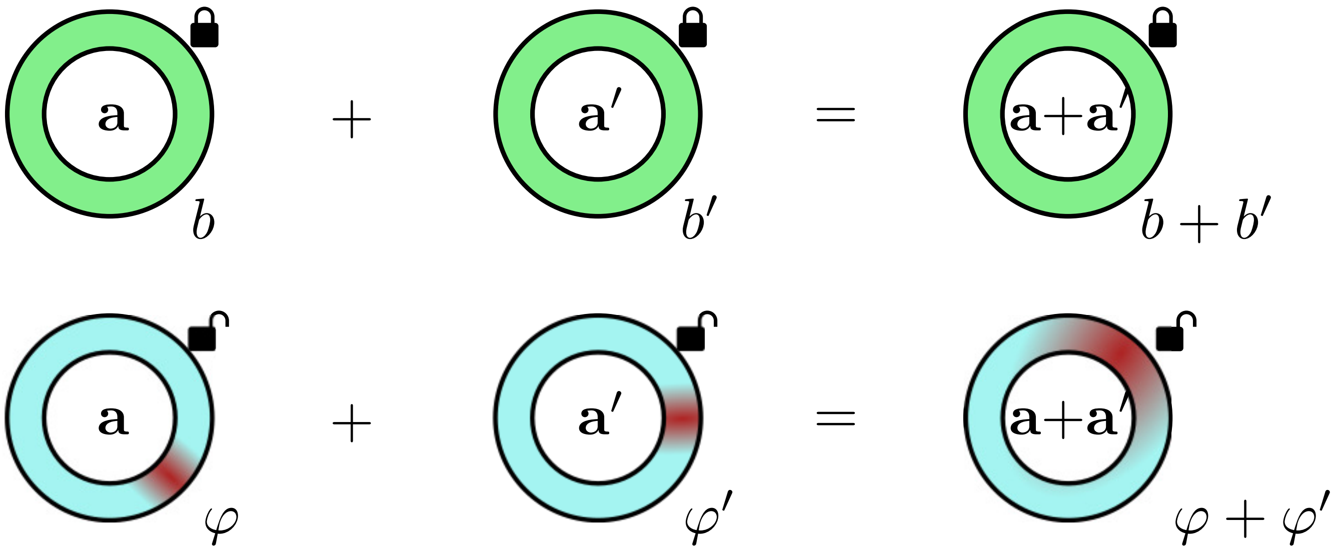
Trivial LWE samples

- LWE samples with mask $\mathbf{a} = \mathbf{0}$ are trivial.
- They never occur in general

...but are still worth mentionning!

LWE

Homomorphic Properties



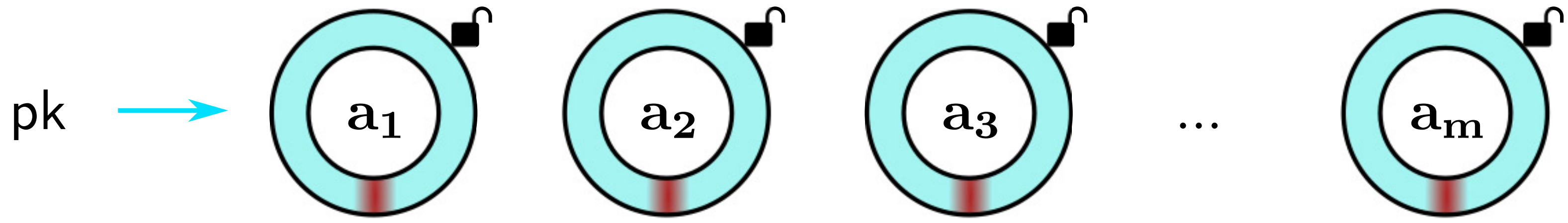
μ $\qquad$ μ' $\qquad$ $\mu'' = \mu + \mu'$

α $\qquad$ α' $\qquad$ $\alpha'' = \sqrt{\alpha^2 + \alpha'^2}$

NAND achieved by composing additions and bootstrapping
→ with NAND we can evaluate every circuit!

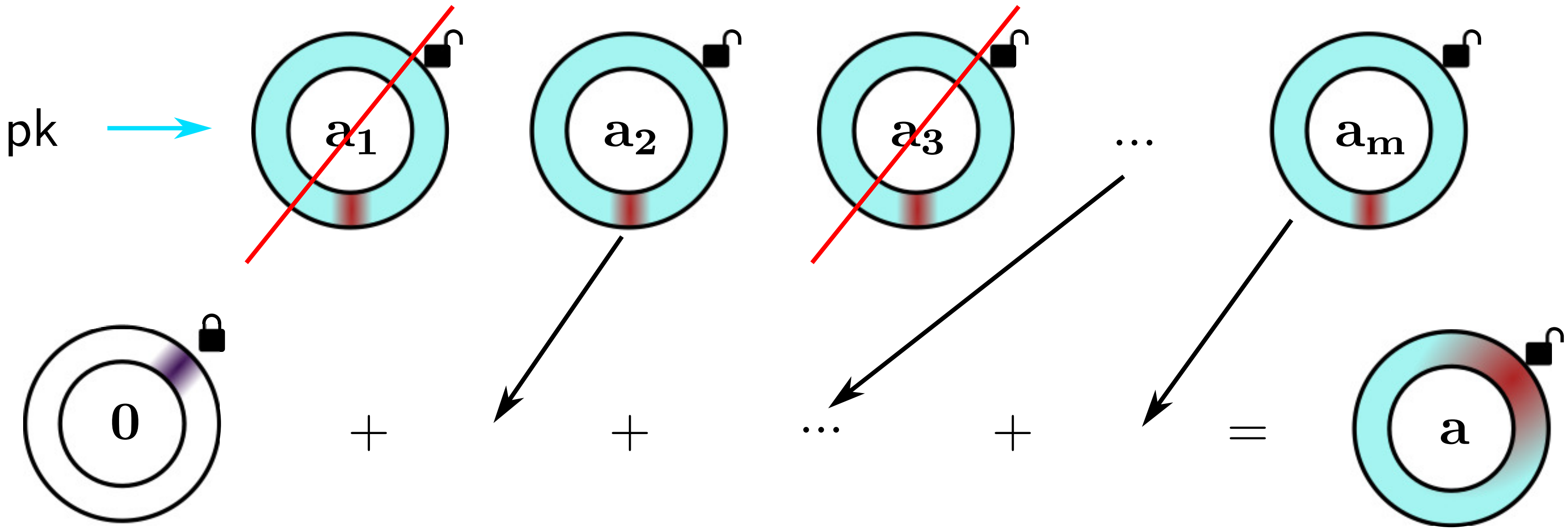
LWE

LWE Asymmetric Encryption



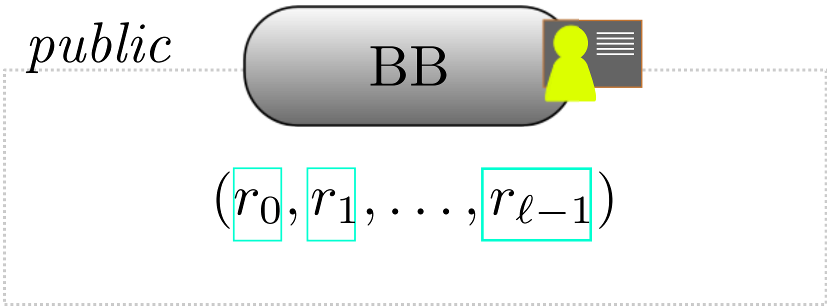
LWE

LWE Asymmetric Encryption



Can be done on the public view, without knowing sk

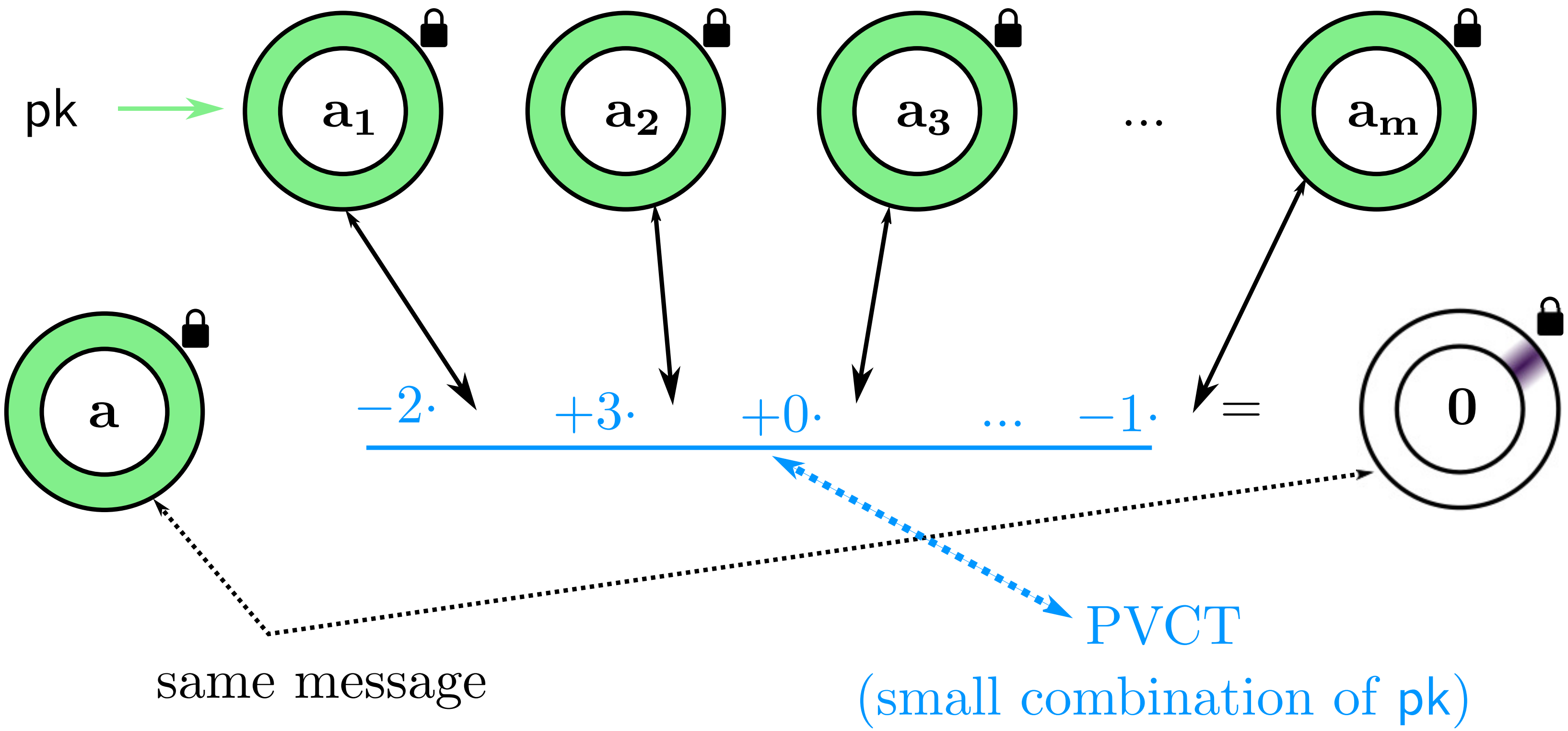
Overview



LWE and PVCT

It is possible to **prove the correct decryption** without revealing any information on the secret?

Publicly Verifiable Ciphertext Trapdoors (PVCT)
Combining [GPV08] and [MP12]



Decrypting the result

The PVCT :

- Does not reveal anything about the secret **s** (using [GPV08])
- Proves the decryption of the ciphertext
- Does not decrypt anything else

Finding a PVCT requires to invert a one-way SIS function

But there is a **trapdoor** solution [MP12]

Setup phase: trustees generate

- Secret keys (**concatenated LWE**)
- The trapdoors

Decrypting the result

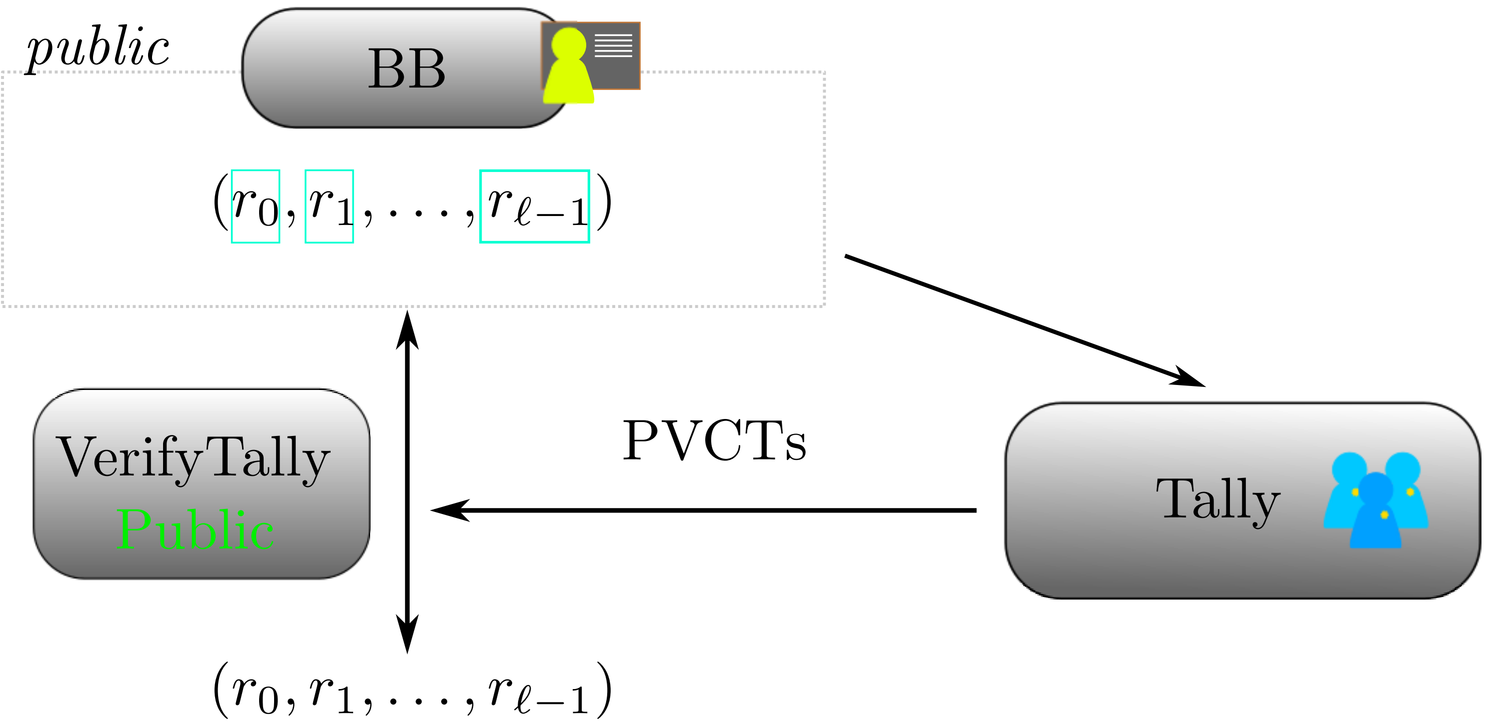
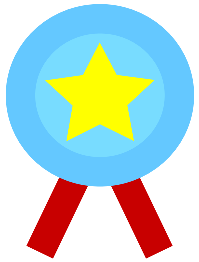


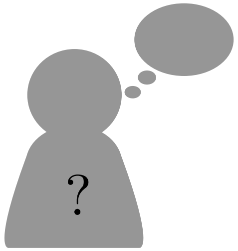
Table of contents

Properties

Correctness

Verifiability



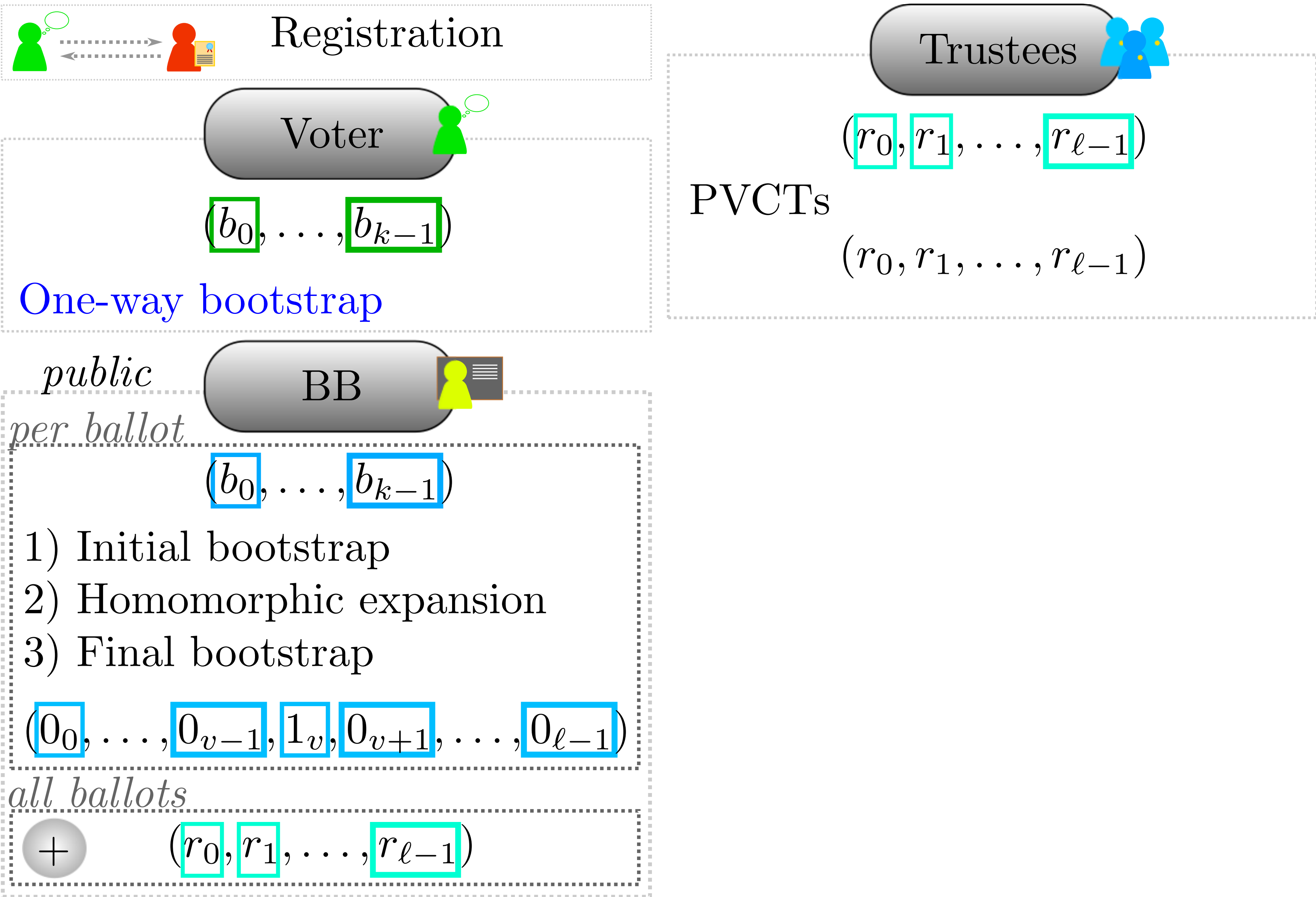


Privacy

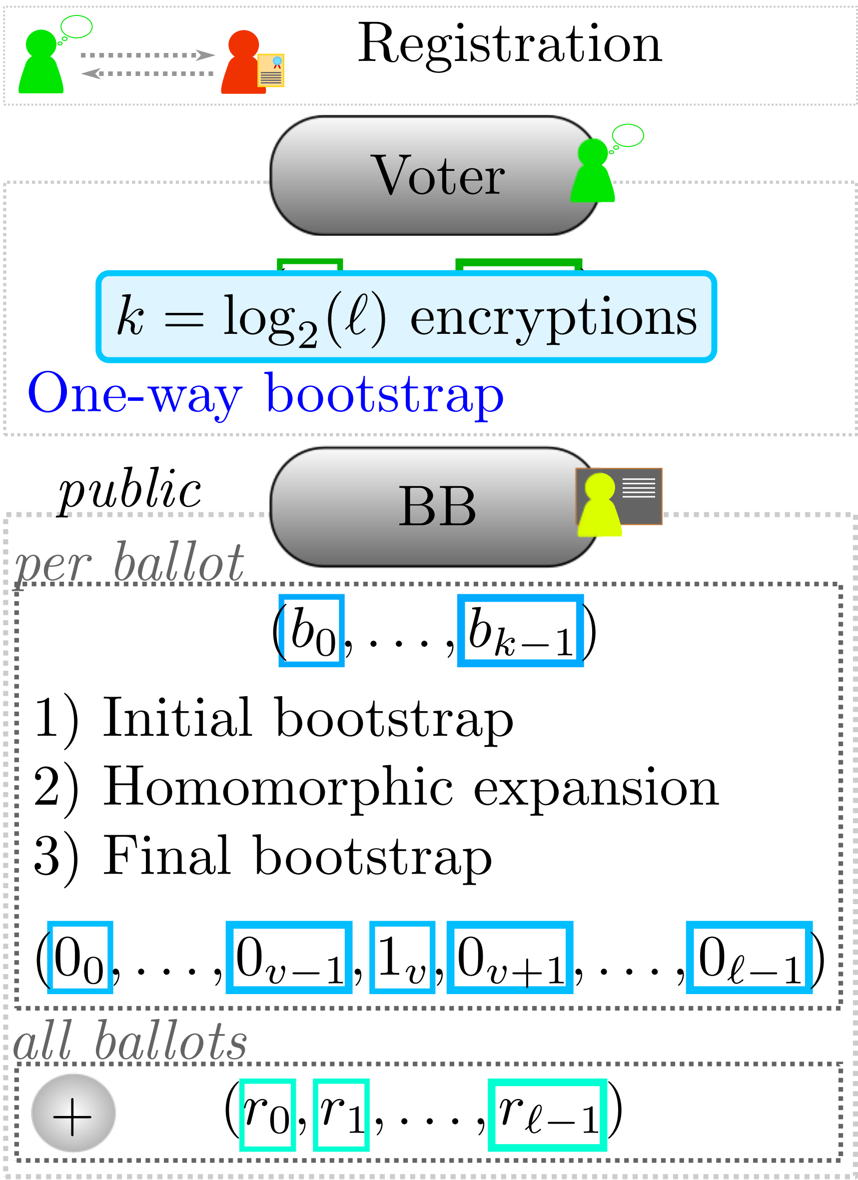
Assumption: *Bootstrapping as a random oracle*

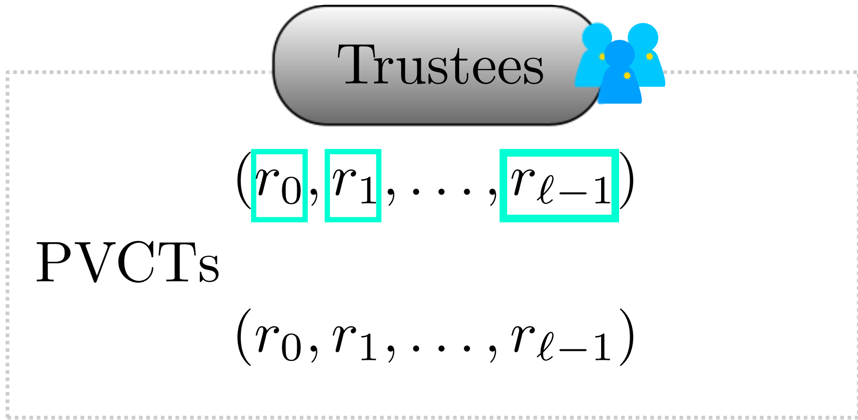
Table of contents

Summary and performance



Summary and performance





1) Voters:
- k bootstraps

2) BB (*per ballot*):
- k bootstraps to verify
- k initial bootstraps
- $(k \times \ell)$ NAND gates
- ℓ *final bootstraps*

Homomorphic sum $\oplus$

3) Trustees (*each*) :
- ℓ generations of PVCTs

Conclusion

	Helios	**Our protocol**
Homomorphic Encrypt	Additive	Fully
Security	DL (*ElGamal*)	Lattice-based (*LWE*)
Secret sharing	Shamir	Concatenated LWE
Honest Trustees	66%	1
Ballot shape	Initial NIZK proof	Full domain
Correct decrypt	Final NIZK proof	PVCT

✓ Any attempt of cheating is publicly detected

Open problems and future work

- Prove the same strong privacy without relying on the assumption
- Additional properties
- Implement the scheme in practice

Arigatō!