Privacy-Preserving Cryptography from Pairings and Lattices

Fabrice Mouhartem
Under the supervision of Benoît Libert

October 18th, 2018

École Normale Supérieure de Lyon, France

Privacy-Preserving Cryptography

Important Goal

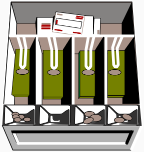
Allowing functionality while preserving anonymity

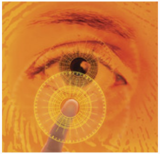
Privacy-Preserving Cryptography

Important Goal

Allowing functionality while preserving anonymity

e.g. e-voting, e-cash, group signatures, group encryption, ...



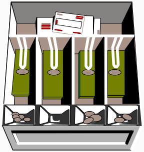


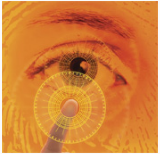
Privacy-Preserving Cryptography

Important Goal

Allowing functionality while preserving anonymity

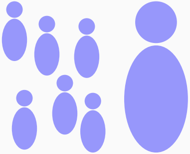
e.g. e-voting, e-cash, group signatures, group encryption, ...





Example: (Dynamic) Group Signatures (Chaum-van Heyst, Eurocrypt'91)

A user wants to take public transportations.



Example: (Dynamic) Group Signatures (Chaum-van Heyst, Eurocrypt'91)

A user wants to take public transportations.

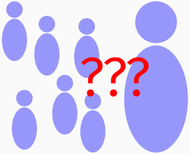
Example: (Dynamic) Group Signatures (Chaum-van Heyst, Eurocrypt'91)

A user wants to take public transportations.

- Authenticity & Integrity

Example: (Dynamic) Group Signatures (Chaum-van Heyst, Eurocrypt'91)

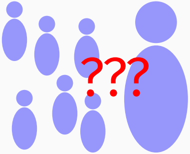
A user wants to take public transportations.



signature →

- Authenticity & Integrity
- Anonymity

A user wants to take public transportations.



signature →

- Authenticity & Integrity
- Anonymity
- Dynamicity ← Join →

Example: (Dynamic) Group Signatures (Chaum-van Heyst, Eurocrypt'91)

A user wants to take public transportations.

- Authenticity & Integrity
- Anonymity
- Dynamicity $\xleftrightarrow{\text{Join}}$
- Traceability

Outline

Practical group signature
(AsiaCCS'16)

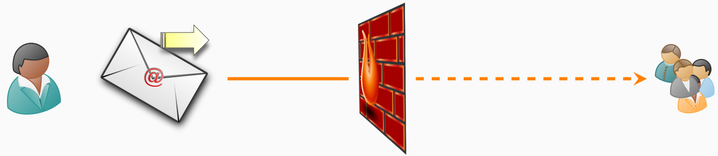
Pairings

First lattice-based signature with efficient protocols
(Asiacrypt'16)

ZK argument of correct evaluation of committed branching programs
(Asiacrypt'16)

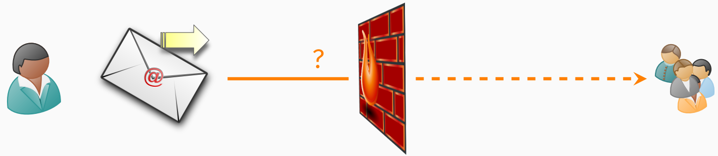
Lattices

ZK argument for quadratic relations
(Asiacrypt'17)



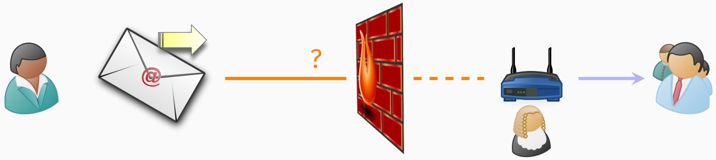
- A user wants to send a message to a group behind a firewall

Motivation: Firewall Filtering



- A user wants to send a message to a group behind a firewall
- The recipient of the message can be a sensitive information

Motivation: Firewall Filtering



- A user wants to send a message to a group behind a firewall
- The recipient of the message can be a sensitive information
- Behind firewall: anonymity is lifted to route messages

History of Group Encryption

2007 Introduction of group encryption (Kiayias-Tsiounis-Yung, Asiacrypt'07)

- Modular design from anonymous PKE, signatures and interactive ZK proofs
- Instantiation using number-theoretic assumptions

History of Group Encryption

2007 Introduction of group encryption (Kiayias-Tsiounis-Yung, Asiacrypt'07)

- Modular design from anonymous PKE, signatures and interactive ZK proofs
- Instantiation using number-theoretic assumptions

2009 Non-interactive GE in the standard model from pairings (Cathalo-Libert-Yung, Asiacrypt'09)

History of Group Encryption

2007 Introduction of group encryption (Kiayias-Tsiounis-Yung, Asiacrypt'07)

- Modular design from anonymous PKE, signatures and interactive ZK proofs
- Instantiation using number-theoretic assumptions

2009 Non-interactive GE in the standard model from pairings (Cathalo-Libert-Yung, Asiacrypt'09)

2013 Various improvements (El Aimani-Joye, ACNS'13)

History of Group Encryption

2007 Introduction of group encryption (Kiayias-Tsiounis-Yung, Asiacrypt'07)

- Modular design from anonymous PKE, signatures and interactive ZK proofs
- Instantiation using number-theoretic assumptions

2009 Non-interactive GE in the standard model from pairings (Cathalo-Libert-Yung, Asiacrypt'09)

2013 Various improvements (El Aimani-Joye, ACNS'13)

2014 Refined traceability mechanism (Libert-Yung-Peters-Joye, PKC'14)

History of Group Encryption

2007 Introduction of group encryption (Kiayias-Tsiounis-Yung, Asiacrypt'07)

- Modular design from anonymous PKE, signatures and interactive ZK proofs
- Instantiation using number-theoretic assumptions

2009 Non-interactive GE in the standard model from pairings (Cathalo-Libert-Yung, Asiacrypt'09)

2013 Various improvements (El Aimani-Joye, ACNS'13)

2014 Refined traceability mechanism (Libert-Yung-Peters-Joye, PKC'14)

✗ Existing realizations rely on quantum-vulnerable assumptions

History of Group Encryption

2007 Introduction of group encryption (Kiayias-Tsiounis-Yung, Asiacrypt'07)

- Modular design from anonymous PKE, signatures and interactive ZK proofs
- Instantiation using number-theoretic assumptions

2009 Non-interactive GE in the standard model from pairings (Cathalo-Libert-Yung, Asiacrypt'09)

2013 Various improvements (El Aimani-Joye, ACNS'13)

2014 Refined traceability mechanism (Libert-Yung-Peters-Joye, PKC'14)

✗ Existing realizations rely on quantum-vulnerable assumptions

→ From lattices: several realizations of group signatures: [GKV10, CNR12, LLLS13, NNZ15, LNW15, LLNW16, LMN16, LLMN16]

Outline

Practical group signature (AsiaCCS'16)

Pairings

First lattice-based signature with efficient protocols (Asiacrypt'16)

ZK argument of correct evaluation of committed branching programs (Asiacrypt'16)

Lattices

ZK argument for quadratic relations (Asiacrypt'17)

Group Encryption (Kyiayias-Tsiounis-Yung, Asiacrypt'07)

Encryption analogue of group signatures:
Sender can encrypt a message to an anonymous group member while proving additional properties.

Group Encryption (Kyiayias-Tsiounis-Yung, Asiacrypt'07)

Encryption analogue of group signatures:
Sender can encrypt a message to an anonymous group member while proving additional properties.

Applications

Firewall filtering, key recovery, anonymous cloud storage, ...

Group Encryption (Kyiayias-Tsiounis-Yung, Asiacrypt'07)

Encryption analogue of group signatures:
Sender can encrypt a message to an anonymous group member while proving additional properties.

Applications

Firewall filtering, key recovery, anonymous cloud storage, ...

Definition

A set of algorithms or protocols: $(\mathsf{Setup}, \mathsf{Join}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Open}, \langle \mathcal{P}, \mathcal{V} \rangle)$

Group Encryption (Kyiayias-Tsiounis-Yung, Asiacrypt'07)

Encryption analogue of group signatures:
Sender can encrypt a message to an anonymous group member while proving additional properties.

Applications

Firewall filtering, key recovery, anonymous cloud storage, ...

Definition

A set of algorithms or protocols: $(\mathsf{Setup}, \mathsf{Join}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Open}, \langle \mathcal{P}, \mathcal{V} \rangle)$

Properties:

Group Encryption (Kyiayias-Tsiounis-Yung, Asiacrypt'07)

Encryption analogue of group signatures:
Sender can encrypt a message to an anonymous group member while proving additional properties.

Applications

Firewall filtering, key recovery, anonymous cloud storage, ...

Definition

A set of algorithms or protocols: $(\mathsf{Setup}, \mathsf{Join}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Open}, \langle \mathcal{P}, \mathcal{V} \rangle)$

Properties:

- Message secrecy

Group Encryption (Kyiayias-Tsiounis-Yung, Asiacrypt'07)

Encryption analogue of group signatures:
Sender can encrypt a message to an anonymous group member while proving additional properties.

Applications

Firewall filtering, key recovery, anonymous cloud storage, ...

Definition

A set of algorithms or protocols: $(\mathsf{Setup}, \mathsf{Join}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Open}, \langle \mathcal{P}, \mathcal{V} \rangle)$

Properties:

- Message secrecy
- Receiver anonymity (within a group)

Group Encryption (Kyiayias-Tsiounis-Yung, Asiacrypt'07)

Encryption analogue of group signatures:
Sender can encrypt a message to an anonymous group member while proving additional properties.

Applications

Firewall filtering, key recovery, anonymous cloud storage, ...

Definition

A set of algorithms or protocols: $(\mathsf{Setup}, \mathsf{Join}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Open}, \langle \mathcal{P}, \mathcal{V} \rangle)$

Properties:

- Message secrecy
- Receiver anonymity (within a group)
- Soundness ($\Rightarrow$ traceability)

Group Encryption (Kyiayias-Tsiounis-Yung, Asiacrypt'07)

Encryption analogue of group signatures:
Sender can encrypt a message to an anonymous group member while proving additional properties.

Applications

Firewall filtering, key recovery, anonymous cloud storage, ...

Definition

A set of algorithms or protocols: $(\mathsf{Setup}, \mathsf{Join}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Open}, \langle \mathcal{P}, \mathcal{V} \rangle)$

Properties:

- Message secrecy
- Receiver anonymity (within a group)
- Soundness ($\Rightarrow$ traceability)

Group Encryption: Receiver Anonymity

Indistinguishability-based game

$\mathcal{A}$

$\mathcal{C}$

Group Encryption: Receiver Anonymity

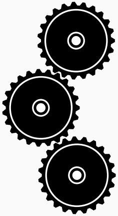
Indistinguishability-based game

$\mathcal{A}$

$\mathcal{C}$

$(\mathsf{pp}, \mathsf{pk}_{\mathrm{OA}}, \mathsf{sk}_{\mathrm{OA}}) \leftarrow \mathsf{Setup}(1^\lambda)$

$\xleftarrow[\mathsf{pp},\, \mathsf{pk}_{\mathrm{OA}}]{}$



Group Encryption: Receiver Anonymity

Indistinguishability-based game

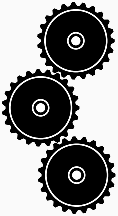
$\mathcal{A}$

$\mathcal{C}$

$(\mathsf{pp}, \mathsf{pk}_{\mathrm{OA}}, \mathsf{sk}_{\mathrm{OA}}) \leftarrow \mathsf{Setup}(1^\lambda)$

$\xleftarrow{\quad \mathsf{pp}, \mathsf{pk}_{\mathrm{OA}} \quad}$

$\xrightarrow{\quad \mathsf{pk}_{\mathrm{GM}} \quad}$



Group Encryption: Receiver Anonymity

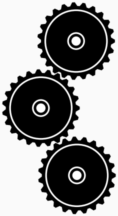
Indistinguishability-based game

$(\mathsf{pp}, \mathsf{pk}_{\mathrm{OA}}, \mathsf{sk}_{\mathrm{OA}}) \leftarrow \mathsf{Setup}(1^\lambda)$

$\xleftarrow{\mathsf{pp}, \mathsf{pk}_{\mathrm{OA}}}$

$\xrightarrow{\mathsf{pk}_{\mathrm{GM}}}$

$\xrightarrow{\mathtt{JOIN}}$



Group Encryption: Receiver Anonymity

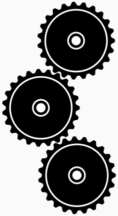
Indistinguishability-based game

$\mathcal{A}$

$\mathcal{C}$

$(\mathrm{pp}, \mathrm{pk}_{\mathrm{OA}}, \mathrm{sk}_{\mathrm{OA}}) \leftarrow \mathrm{Setup}(1^\lambda)$

$\xleftarrow{\quad \mathrm{pp}, \mathrm{pk}_{\mathrm{OA}} \quad}$



$\xrightarrow{\quad \mathrm{pk}_{\mathrm{GM}} \quad}$

$\xrightarrow{\quad \mathtt{JOIN} \quad}$

Simulates Join protocol to obtain $\mathrm{pk}_0, \mathrm{sk}_0, \sigma_0, \mathrm{pk}_1, \mathrm{sk}_1, \sigma_1$

Group Encryption: Receiver Anonymity

Indistinguishability-based game

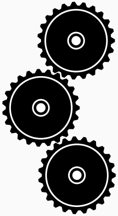
$\mathcal{A}$

$\mathcal{C}$

$(\mathsf{pp}, \mathsf{pk}_{\mathrm{OA}}, \mathsf{sk}_{\mathrm{OA}}) \leftarrow \mathsf{Setup}(1^{\lambda})$

$\xleftarrow{\quad \mathsf{pp}, \mathsf{pk}_{\mathrm{OA}} \quad}$

$\xrightarrow{\quad \mathsf{pk}_{\mathrm{GM}} \quad}$

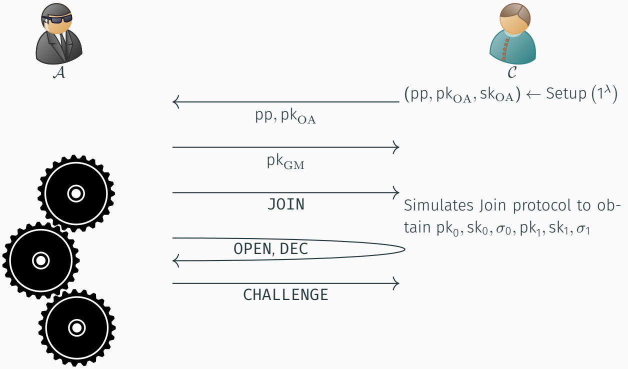


$\xrightarrow{\quad \texttt{JOIN} \quad}$

Simulates Join protocol to obtain $\mathsf{pk}_0, \mathsf{sk}_0, \sigma_0, \mathsf{pk}_1, \mathsf{sk}_1, \sigma_1$

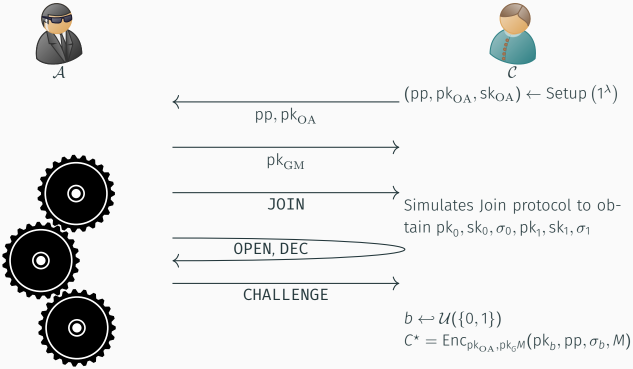
$\xleftarrow{\quad \texttt{OPEN, DEC} \quad}$

Indistinguishability-based game



Group Encryption: Receiver Anonymity

Indistinguishability-based game

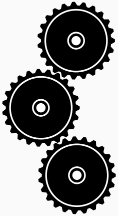


Group Encryption: Receiver Anonymity

Indistinguishability-based game

$\mathcal{A}$

$\mathcal{C}$



$(\mathrm{pp}, \mathrm{pk}_{\mathrm{OA}}, \mathrm{sk}_{\mathrm{OA}}) \leftarrow \mathsf{Setup}(1^\lambda)$

$\xleftarrow{\mathrm{pp}, \mathrm{pk}_{\mathrm{OA}}}$

$\xrightarrow{\mathrm{pk}_{\mathrm{GM}}}$

$\xrightarrow{\mathtt{JOIN}}$

Simulates Join protocol to obtain $\mathrm{pk}_0, \mathrm{sk}_0, \sigma_0, \mathrm{pk}_1, \mathrm{sk}_1, \sigma_1$

OPEN, DEC

$\xrightarrow{\mathtt{CHALLENGE}}$

$b \hookleftarrow \mathcal{U}(\{0,1\})$

$C^\star = \mathsf{Enc}_{\mathrm{pk}_{\mathrm{OA}}, \mathrm{pk}_{GM}}(\mathrm{pk}_b, \mathrm{pp}, \sigma_b, M)$

$\xleftarrow{C^\star}$

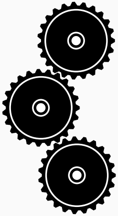
OPEN*, DEC*

Group Encryption: Receiver Anonymity

Indistinguishability-based game

$\mathcal{A}$

$\mathcal{C}$



$(\mathsf{pp}, \mathsf{pk}_{\mathrm{OA}}, \mathsf{sk}_{\mathrm{OA}}) \leftarrow \mathsf{Setup}(1^\lambda)$

$\xleftarrow{\mathsf{pp}, \mathsf{pk}_{\mathrm{OA}}}$

$\xrightarrow{\mathsf{pk}_{\mathrm{GM}}}$

$\xrightarrow{\texttt{JOIN}}$

Simulates Join protocol to obtain $\mathsf{pk}_0, \mathsf{sk}_0, \sigma_0, \mathsf{pk}_1, \mathsf{sk}_1, \sigma_1$

OPEN, DEC

$\xrightarrow{\texttt{CHALLENGE}}$

$b \hookleftarrow \mathcal{U}(\{0,1\})$

$C^\star = \mathsf{Enc}_{\mathsf{pk}_{\mathrm{OA}}, \mathsf{pk}_G M}(\mathsf{pk}_b, \mathsf{pp}, \sigma_b, M)$

$\xleftarrow{C^\star}$

OPEN*, DEC*

$\xrightarrow{b'}$

Indistinguishability-based game

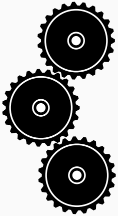
$\mathcal{A}$

$\mathcal{C}$

$(\mathsf{pp}, \mathsf{pk}_{\mathrm{OA}}, \mathsf{sk}_{\mathrm{OA}}) \leftarrow \mathsf{Setup}(1^\lambda)$

$\xleftarrow{\mathsf{pp}, \mathsf{pk}_{\mathrm{OA}}}$

$\xrightarrow{\mathsf{pk}_{\mathrm{GM}}}$



$\xrightarrow{\texttt{JOIN}}$

Simulates Join protocol to obtain $\mathsf{pk}_0, \mathsf{sk}_0, \sigma_0, \mathsf{pk}_1, \mathsf{sk}_1, \sigma_1$

OPEN, DEC

$\xrightarrow{\texttt{CHALLENGE}}$

$b \hookleftarrow \mathcal{U}(\{0,1\})$

$C^\star = \mathsf{Enc}_{\mathsf{pk}_{\mathrm{OA}}, \mathsf{pk}_{GM}}(\mathsf{pk}_b, \mathsf{pp}, \sigma_b, M)$

$\xleftarrow{C^\star}$

OPEN*, DEC*

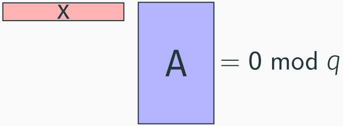
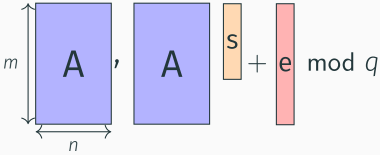
$\xrightarrow{b'}$

$\mathcal{A}$ wins if $b = b'$

Hardness Assumptions: SIS and LWE (Ajtai 1996, Regev 2005)

Parameters: dimension n, #samples $m \geq n$, modulus q.

For $\boxed{A} \hookleftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$:

Small Integer Solution	Learning With Errors
	
	$\boxed{s} \hookleftarrow \mathbb{Z}_q^n$ $\quad$ $\boxed{e}$ small error
Goal: Given $\boxed{A} \in \mathbb{Z}_q^{m \times n}$, find $\boxed{x} \in \mathbb{Z}^m \setminus \{0\}$ small	Goal: Given $\boxed{A}$, $\boxed{A}\boxed{s} + \boxed{e}$, find $\boxed{s} \in \mathbb{Z}_q^n$

Lattice-Based Cryptography (Ajtai 1996, Regev 2005)

Why?

- Simple and asymptotically efficient;
- Conjectured quantum-resistant;
- Connection between average-case and worst-case problems;
- Powerful functionalities (e.g., FHE).

Lattice-Based Cryptography (Ajtai 1996, Regev 2005)

Why?

- Simple and asymptotically efficient;
- Conjectured quantum-resistant;
- Connection between average-case and worst-case problems;
- Powerful functionalities (e.g., FHE).

Remark: GS and GE rely on the same building blocks:

- Digital signatures;
- Public-Key encryption;
- Supporting Zero-Knowledge proofs.

Lattice-Based Cryptography (Ajtai 1996, Regev 2005)

Why?

- Simple and asymptotically efficient;
- Conjectured quantum-resistant;
- Connection between average-case and worst-case problems;
- Powerful functionalities (e.g., FHE).

Remark: GS and GE rely on the same building blocks:

- Digital signatures;
- Public-Key encryption;
- Supporting Zero-Knowledge proofs.

What is the main difficulty?

Interactive protocol between prover P and verifier V such that:

Completeness: Correctness of the protocol.

Zero-Knowledge Proofs (Goldwasser-Micali-Rackoff, STOC'85)

Interactive protocol between prover P and verifier V such that:

Completeness: Correctness of the protocol.

Soundness: No cheating prover can convince the verifier.

Zero-Knowledge Proofs (Goldwasser-Micali-Rackoff, STOC'85)

Interactive protocol between prover P and verifier V such that:

Completeness: Correctness of the protocol.

Soundness: No cheating prover can convince the verifier.

Zero-Knowledge: Verifier learns nothing but the validity of the statement.

Zero-Knowledge Proofs (Goldwasser-Micali-Rackoff, STOC'85)

Interactive protocol between prover P and verifier V such that:

Completeness: Correctness of the protocol.

Soundness: No cheating prover can convince the verifier.

Zero-Knowledge: Verifier learns nothing but the validity of the statement.

- Non-interactive variants: NIZK proofs
- Random Oracle: allows transforming ZK to NIZK (Fiat-Shamir, Crypto'86)
- Standard Model: using bilinear maps (Groth-Sahai, Eurocrypt'08)

Zero-Knowledge Proofs for Lattices

Two main proof systems in lattice-based cryptography:

Schnorr-like (Crypto'89): On Ring-LWE[1], concise but not expressive.

Stern-like (Crypto'93): On LWE[2], heavy but expressive.

[1] Lyubashevsky, Asiacrypt'09

[2] Kawachi-Tanaka-Xagawa, Asiacrypt'08 and Ling-Nguyen-Stehlé-Wang, PKC'13

Two main proof systems in lattice-based cryptography:

Schnorr-like (Crypto'89): On Ring-LWE[1], concise but not expressive.

Stern-like (Crypto'93): On LWE[2], heavy but expressive.

Both deal with "linear relations", i.e., of the form

$$\boxed{\mathrm{X}} \cdot \boxed{\mathrm{s}} = \boxed{\mathrm{y}} \bmod q$$

[1] Lyubashevsky, Asiacrypt'09

[2] Kawachi-Tanaka-Xagawa, Asiacrypt'08 and Ling-Nguyen-Stehlé-Wang, PKC'13

Zero-Knowledge Proofs for Lattices

Two main proof systems in lattice-based cryptography:

Schnorr-like (Crypto'89): On Ring-LWE[1], concise but not expressive.

Stern-like (Crypto'93): On LWE[2], heavy but expressive.

Both deal with "linear relations", i.e., of the form

$$\boxed{X} \cdot \boxed{s} = \boxed{y} \bmod q$$

Examples: (I)SIS and LWE relations are linear

[1]Lyubashevsky, Asiacrypt'09

[2]Kawachi-Tanaka-Xagawa, Asiacrypt'08 and Ling-Nguyen-Stehlé-Wang, PKC'13

The Case of Group Signatures

Modular design for GS[1]: sign-then-encrypt-then-prove

- GM issues a signature σ on *id* to each user

[1] Bellare, Micciancio and Warinschi at Eurocrypt'03

The Case of Group Signatures

Modular design for GS[1]: **sign-then-encrypt-then-prove**

- GM issues a signature σ on *id* to each user
- Sign:
  - a user encrypts *id* to c under OA's public key $\mathsf{pk}_{\mathrm{OA}}$

[1] Bellare, Micciancio and Warinschi at Eurocrypt'03

The Case of Group Signatures

Modular design for GS[1]: sign-then-encrypt-then-prove

- GM issues a signature σ on *id* to each user
- Sign:
  - a user encrypts *id* to c under OA's public key $\mathsf{pk}_{\mathrm{OA}}$
  - User proves that:
    1. He has a secret valid pair (id, σ), w.r.t. $\mathsf{vk}_{\mathrm{GM}}$
    2. c is a valid encryption of *id*, w.r.t. $\mathsf{pk}_{\mathrm{OA}}$

[1] Bellare, Micciancio and Warinschi at Eurocrypt'03

The Case of Group Signatures

Modular design for GS[1]: **sign-then-encrypt-then-prove**

- GM issues a signature σ on *id* to each user
- Sign:
  - a user encrypts *id* to c under OA's public key $\mathsf{pk}_{\mathrm{OA}}$
  - User proves that:
    1. He has a secret valid pair (id, σ), w.r.t. $\mathsf{vk}_{\mathrm{GM}}$ ISIS
    2. c is a valid encryption of *id*, w.r.t. $\mathsf{pk}_{\mathrm{OA}}$ LWE

✓ Known techniques allow realizing the ZK proofs

[1] Bellare, Micciancio and Warinschi at Eurocrypt'03

The Case of Group Signatures

Modular design for GS[1]: **sign-then-encrypt-then-prove**

- GM issues a signature σ on *id* to each user
- **Sign**:
  - a user encrypts *id* to c under OA's public key $\mathsf{pk}_{\mathrm{OA}}$
  - User proves that:
    1. He has a secret valid pair (id, σ), w.r.t. $\mathsf{vk}_{\mathrm{GM}}$ — ISIS
    2. c is a valid encryption of *id*, w.r.t. $\mathsf{pk}_{\mathrm{OA}}$ — LWE

✓ Known techniques allow realizing the ZK proofs

Remark: The message is embedded in the NIZK proof

[1] Bellare, Micciancio and Warinschi at Eurocrypt'03

Main Difficulty in Group Encryption

Modular design (Kiayias-Tsiounis-Yung, Asiacrypt'07):

- Each member has an anonymous encryption key pair $(\mathsf{pk}, \mathsf{sk})$
- GM signs each pk and publishes (pk, σ)

Main Difficulty in Group Encryption

Modular design (Kiayias-Tsiounis-Yung, Asiacrypt'07):

- Each member has an anonymous encryption key pair $(\mathsf{pk}, \mathsf{sk})$
- GM signs each pk and publishes (pk, σ)
- Sender uses pk to encrypt a message μ satisfying $\mathcal{R}$; obtains c
- Sender also encrypts pk under $\mathsf{pk}_{\mathrm{OA}}$, obtains c_{OA}

Main Difficulty in Group Encryption

Modular design (Kiayias-Tsiounis-Yung, Asiacrypt'07):

- Each member has an anonymous encryption key pair $(\mathsf{pk}, \mathsf{sk})$
- GM signs each pk and publishes (pk, σ)
- Sender uses pk to encrypt a message μ satisfying $\mathcal{R}$; obtains c
- Sender also encrypts pk under $\mathsf{pk}_{\mathrm{OA}}$, obtains c_{OA}
- Sender proves:
  1. $\mathsf{c} = \mathsf{Enc}_{\mathsf{pk}}(\mu)$
  2. Knowledge of σ s.t. $\mathsf{Verif}_{\mathsf{vk}_{\mathrm{GM}}}(\mathsf{pk}, \sigma)$; $\mathsf{c}_{\mathrm{OA}} = \mathsf{Enc}_{\mathsf{pk}_{\mathrm{OA}}}(\mathsf{pk})$; $\mathcal{R}(\mu) = \top$.

Main Difficulty in Group Encryption

Modular design (Kiayias-Tsiounis-Yung, Asiacrypt'07):

- Each member has an anonymous encryption key pair $(\mathsf{pk}, \mathsf{sk})$
- GM signs each pk and publishes (pk, σ)
- Sender uses pk to encrypt a message μ satisfying $\mathcal{R}$; obtains c
- Sender also encrypts pk under $\mathsf{pk}_{\mathrm{OA}}$, obtains c_{OA}
- Sender proves:
  1. $\mathsf{c} = \mathsf{Enc}_{\mathsf{pk}}(\mu)$
  2. Knowledge of σ s.t. $\mathsf{Verif}_{\mathsf{vk}_{\mathrm{GM}}}(\mathsf{pk}, \sigma)$; $\mathsf{c}_{\mathrm{OA}} = \mathsf{Enc}_{\mathsf{pk}_{\mathrm{OA}}}(\mathsf{pk})$; $\mathcal{R}(\mu) = \top$.

✗ We have to handle relations with **hidden-but-certified** matrix:

$$\boxed{\mathsf{X}} \cdot \boxed{\mathsf{s}} + \boxed{\mathsf{e}} = \boxed{\mathsf{b}} \bmod q$$

Stern's Protocol (Crypto'93)

Stern's protocol is a ZK proof for Syndrome Decoding Problem.

Stern's protocol is a ZK proof for Syndrome Decoding Problem.

Syndrome Decoding Problem

Given $\boxed{P} \in \mathbb{Z}_2^{n\times m}$ and $\boxed{v} \in \mathbb{Z}_2^{n}$, find $\boxed{x}$ s.t. $\mathrm{w}(\boxed{x}) = w$ and

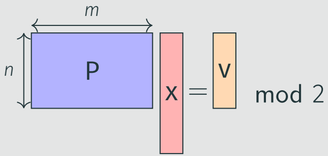
$$n \updownarrow \overbrace{\boxed{\quad P \quad}}^{m} \, \boxed{x} = \boxed{v} \mod 2$$

Stern's Protocol (Crypto'93)

Stern's protocol is a ZK proof for Syndrome Decoding Problem.

Syndrome Decoding Problem

Given $\boxed{P} \in \mathbb{Z}_2^{n \times m}$ and $\boxed{v} \in \mathbb{Z}_2^{n}$, find $\boxed{x}$ s.t. $\mathrm{w}(\boxed{x}) = w$ and



Kawachi-Tanaka-Xagawa'08: mod $2 \to$ mod q

Ling-Nguyen-Stehlé-Wang'13: Extends Stern's protocol to SIS/LWE

Recent uses of Stern-like protocols in lattice-based crypto:
[LNW15, LLNW16, LMN16, LLNMW16, LLNMW17, LLNW17]

Stern's Ideas

Syndrome Decoding Problem

Given $\boxed{\mathsf{P}} \in \mathbb{Z}_2^{n \times m}$ and $\boxed{\mathsf{v}} \in \mathbb{Z}_2^{n}$, find $\boxed{\mathsf{x}}$ s.t. $\mathrm{w}(\boxed{\mathsf{x}}) = w$ and

$$\boxed{\mathsf{P}} \cdot \boxed{\mathsf{x}} = \boxed{\mathsf{v}} \bmod 2$$

Stern's Ideas

Syndrome Decoding Problem

Given $\boxed{\mathsf{P}} \in \mathbb{Z}_2^{n\times m}$ and $\boxed{\mathsf{v}} \in \mathbb{Z}_2^n$, find $\boxed{\mathsf{x}}$ s.t. $\mathrm{w}(\boxed{\mathsf{x}}) = w$ and

$$\boxed{\mathsf{P}} \cdot \boxed{\mathsf{x}} = \boxed{\mathsf{v}} \bmod 2$$

1. **Permuting:** Random permutation proves constraints on $\boxed{\mathsf{x}}$
   - Send the verifier $\pi(\boxed{\mathsf{x}})$
   - $\boxed{\mathsf{x}}$ binary of hamming weight $w \Leftrightarrow \pi(\boxed{\mathsf{x}})$ does
   - ☞ π's randomness preserves the secrecy of $\boxed{\mathsf{x}}$

Stern's Ideas

Syndrome Decoding Problem

Given $\boxed{\mathsf{P}} \in \mathbb{Z}_2^{n \times m}$ and $\boxed{\mathsf{v}} \in \mathbb{Z}_2^n$, find $\boxed{\mathsf{x}}$ s.t. $\mathrm{w}(\boxed{\mathsf{x}}) = w$ and

$$\boxed{\mathsf{P}} \cdot \boxed{\mathsf{x}} = \boxed{\mathsf{v}} \bmod 2$$

1. **Permuting:** Random permutation proves constraints on $\boxed{\mathsf{x}}$
   - Send the verifier $\pi(\boxed{\mathsf{x}})$
   - $\boxed{\mathsf{x}}$ binary of hamming weight $w \Leftrightarrow \pi(\boxed{\mathsf{x}})$ does
   - ☞ π's randomness preserves the secrecy of $\boxed{\mathsf{x}}$
2. **Masking:** Random mask $\boxed{\mathsf{r}}$ is used to prove the linear equation
   - Send the verifier $\boxed{\mathsf{y}} = \boxed{\mathsf{x}} + \boxed{\mathsf{r}}$ and show that $\boxed{\mathsf{P}} \cdot \boxed{\mathsf{y}} = \boxed{\mathsf{v}} + \boxed{\mathsf{P}} \cdot \boxed{\mathsf{r}}$

Stern's Ideas

Syndrome Decoding Problem

Given $\boxed{\mathsf{P}} \in \mathbb{Z}_2^{n \times m}$ and $\boxed{\mathsf{v}} \in \mathbb{Z}_2^n$, find $\boxed{\mathsf{x}}$ s.t. $\mathrm{w}(\boxed{\mathsf{x}}) = w$ and

$$\boxed{\mathsf{P}} \cdot \boxed{\mathsf{x}} = \boxed{\mathsf{v}} \bmod 2$$

1. Permuting: Random permutation proves constraints on $\boxed{\mathsf{x}}$
   - Send the verifier $\pi(\boxed{\mathsf{x}})$
   - $\boxed{\mathsf{x}}$ binary of hamming weight $w \Leftrightarrow \pi(\boxed{\mathsf{x}})$ does
   - ☞ π's randomness preserves the secrecy of $\boxed{\mathsf{x}}$
2. **Masking:** Random mask $\boxed{\mathsf{r}}$ is used to prove the linear equation
   - Send the verifier $\boxed{\mathsf{y}} = \boxed{\mathsf{x}} + \boxed{\mathsf{r}}$ and show that $\boxed{\mathsf{P}} \cdot \boxed{\mathsf{y}} = \boxed{\mathsf{v}} + \boxed{\mathsf{P}} \cdot \boxed{\mathsf{r}}$

Idea:

1. Pre-process the given quadratic relation
2. Exploit permutations to prove the relation

Deal with Quadratic Relations: First Step

Goal: Express $\boxed{X} \cdot \boxed{s}$ as $\boxed{Q} \cdot \boxed{z}$

Deal with Quadratic Relations: First Step

Goal: Express $\boxed{X} \cdot \boxed{s}$ as $\boxed{Q} \cdot \boxed{z}$

Idea: Binary decomposition

1. $\mathbf{X} \cdot \mathbf{s} = \sum_{i=1}^{n} \mathbf{x}_i \cdot s_i$

   $\mathbf{x}_i$ denotes i-th column of $\mathbf{X}$; s_i is the i-th component of $\mathbf{s}$

Deal with Quadratic Relations: First Step

Goal: Express $\boxed{X} \cdot \boxed{s}$ as $\boxed{Q} \cdot \boxed{z}$

Idea: Binary decomposition

1. $\mathbf{X} \cdot \mathbf{s} = \sum_{i=1}^{n} \mathbf{x}_i \cdot s_i$

   $\mathbf{x}_i$ denotes i-th column of $\mathbf{X}$; s_i is the i-th component of $\mathbf{s}$

2. $\mathbf{x}_i \cdot s_i = \mathbf{H} \cdot (x_{i,1} \cdot s_i, \ldots, x_{i,mk} \cdot s_i)^T$

   $k = \lceil \log q \rceil$; $\mathbf{H}$ is the decomposition matrix $\mathbb{Z}_q^m \to \{0,1\}^{mk}$

Deal with Quadratic Relations: First Step

Goal: Express $\boxed{\mathbf{X}} \cdot \boxed{\mathbf{s}}$ as $\boxed{\mathbf{Q}} \cdot \boxed{\mathbf{z}}$

Idea: Binary decomposition

1. $\mathbf{X} \cdot \mathbf{s} = \sum_{i=1}^{n} \mathbf{x}_i \cdot s_i$

   $\mathbf{x}_i$ denotes i-th column of $\mathbf{X}$; s_i is the i-th component of $\mathbf{s}$

2. $\mathbf{x}_i \cdot s_i = \mathbf{H} \cdot (x_{i,1} \cdot s_i, \ldots, x_{i,mk} \cdot s_i)^T$

   $k = \lceil \log q \rceil$; $\mathbf{H}$ is the decomposition matrix $\mathbb{Z}_q^m \to \{0,1\}^{mk}$

3. $$\begin{aligned} x_{i,j} \cdot s_i &= x_{i,j} \cdot (\tilde{h}_1, \ldots, \tilde{h}_k) \cdot (s_{i,1}, \ldots, s_{i,k})^T \\ &= (\tilde{h}_1, \ldots, \tilde{h}_k) \cdot (x_{i,j} \cdot s_{i,1}, \ldots, x_{i,j} \cdot s_{i,k})^T. \end{aligned}$$

Deal with Quadratic Relations: First Step

Goal: Express $\boxed{\mathbf{X}} \cdot \boxed{\mathbf{s}}$ as $\boxed{\mathbf{Q}} \cdot \boxed{\mathbf{z}}$

Idea: Binary decomposition

1. $\mathbf{X} \cdot \mathbf{s} = \sum_{i=1}^{n} \mathbf{x}_i \cdot s_i$

   $\mathbf{x}_i$ denotes i-th column of $\mathbf{X}$; s_i is the i-th component of $\mathbf{s}$

2. $\mathbf{x}_i \cdot s_i = \mathbf{H} \cdot (x_{i,1} \cdot s_i, \ldots, x_{i,mk} \cdot s_i)^T$

   $k = \lceil \log q \rceil$; $\mathbf{H}$ is the decomposition matrix $\mathbb{Z}_q^m \to \{0,1\}^{mk}$

3. $$\begin{aligned} x_{i,j} \cdot s_i &= x_{i,j} \cdot (\tilde{h}_1, \ldots, \tilde{h}_k) \cdot (s_{i,1}, \ldots, s_{i,k})^T \\ &= (\tilde{h}_1, \ldots, \tilde{h}_k) \cdot (x_{i,j} \cdot s_{i,1}, \ldots, x_{i,j} \cdot s_{i,k})^T. \end{aligned}$$

$x_{i,j} \cdot s_i$ has form "(public matrix)·(secret vector)" $\Rightarrow$ so does $\mathbf{x}_i \cdot s_i$

$\Rightarrow$ so does $\boxed{\mathbf{X}} \cdot \boxed{\mathbf{s}} = \boxed{\mathbf{Q}} \cdot \boxed{\mathbf{z}} \bmod q$

Where are we?

We expressed $\boxed{X} \cdot \boxed{s}$ as $\boxed{Q} \cdot \boxed{z}$.

Where are we?

We expressed $\boxed{X} \cdot \boxed{s}$ as $\boxed{Q} \cdot \boxed{z}$.

- $\boxed{z}$ is binary and **quadratic**: each z_i is a product of a bit from $\boxed{X}$ with a bit from $\boxed{s}$

Where are we?

We expressed $\boxed{X} \cdot \boxed{s}$ as $\boxed{Q} \cdot \boxed{z}$.

- $\boxed{z}$ is binary and **quadratic**: each z_i is a product of a bit from $\boxed{X}$ with a bit from $\boxed{s}$
- The component bits additionally satisfy other relations

Where are we?

We expressed $\boxed{X} \cdot \boxed{s}$ as $\boxed{Q} \cdot \boxed{z}$.

- $\boxed{z}$ is binary and **quadratic**: each z_i is a product of a bit from $\boxed{X}$ with a bit from $\boxed{s}$
- The component bits additionally satisfy other relations

Goal

Prove that a secret bit z is of form $z = c_1 \cdot c_2$, while remaining able to prove that the c_1 and c_2 satisfy other equations.

Deal with Quadratic Relations: Second Step

Idea: Two-bit-based permutations

Deal with Quadratic Relations: Second Step

Idea: Two-bit-based permutations

- For $c \in \{0,1\}$ let $\bar{c} = 1 - c$. For $c_1, c_2 \in \{0,1\}$, define

$$\mathrm{ext}(c_1, c_2) = (\bar{c}_1 \cdot \bar{c}_2, \bar{c}_1 \cdot c_2, c_1 \cdot \bar{c}_2, c_1 \cdot c_2)^T$$

Deal with Quadratic Relations: Second Step

Idea: Two-bit-based permutations

- For $c \in \{0,1\}$ let $\bar{c} = 1 - c$. For $c_1, c_2 \in \{0,1\}$, define

$$\mathrm{ext}(c_1, c_2) = (\bar{c}_1 \cdot \bar{c}_2, \bar{c}_1 \cdot c_2, c_1 \cdot \bar{c}_2, c_1 \cdot c_2)^T$$

- For $b_1, b_2 \in \{0,1\}$, define the permutation T_{b_1,b_2}:

$$T_{b_1,b_2}\left((v_{0,0}, v_{0,1}, v_{1,0}, v_{1,1})^T\right) = (v_{b_1,b_2}, v_{b_1,\bar{b}_2}, v_{\bar{b}_1,b_2}, v_{\bar{b}_1,\bar{b}_2})^T$$

Deal with Quadratic Relations: Second Step

Idea: Two-bit-based permutations

- For $c \in \{0,1\}$ let $\bar{c} = 1 - c$. For $c_1, c_2 \in \{0,1\}$, define

$$\mathrm{ext}(c_1, c_2) = (\bar{c}_1 \cdot \bar{c}_2, \bar{c}_1 \cdot c_2, c_1 \cdot \bar{c}_2, c_1 \cdot c_2)^T$$

- For $b_1, b_2 \in \{0,1\}$, define the permutation T_{b_1,b_2}:

$$T_{b_1,b_2}\left((v_{0,0}, v_{0,1}, v_{1,0}, v_{1,1})^T\right) = (v_{b_1,b_2}, v_{b_1,\bar{b}_2}, v_{\bar{b}_1,b_2}, v_{\bar{b}_1,\bar{b}_2})^T$$

Note that for all $c_1, c_2, b_1, b_2 \in \{0,1\}$, it holds that

$$\mathbf{v} = \mathrm{ext}(c_1, c_2) \iff T_{b_1,b_2}(\mathbf{v}) = \mathrm{ext}(c_1 \oplus b_1, c_2 \oplus b_2)$$

Solution to the Sub-Problem

Goal

Prove that a secret bit z is of form $z = c_1 \cdot c_2$, while remaining able to prove that the c_1 and c_2 satisfy other equations.

$$\mathsf{v} = \mathrm{ext}(c_1, c_2) \iff T_{b_1,b_2}(\mathsf{v}) = \mathrm{ext}(c_1 \oplus b_1, c_2 \oplus b_2)$$

Solution to the Sub-Problem

Goal

Prove that a secret bit z is of form $z = c_1 \cdot c_2$, while remaining able to prove that the c_1 and c_2 satisfy other equations.

$$\mathsf{v} = \mathrm{ext}(c_1, c_2) \iff T_{b_1,b_2}(\mathsf{v}) = \mathrm{ext}(c_1 \oplus b_1, c_2 \oplus b_2)$$

- Extend $z = c_1 \cdot c_2$ to $\mathsf{v} = \mathrm{ext}(c_1, c_2)$

Solution to the Sub-Problem

Goal

Prove that a secret bit z is of form $z = c_1 \cdot c_2$, while remaining able to prove that the c_1 and c_2 satisfy other equations.

$$\mathsf{v} = \mathrm{ext}(c_1, c_2) \iff T_{b_1,b_2}(\mathsf{v}) = \mathrm{ext}(c_1 \oplus b_1, c_2 \oplus b_2)$$

- Extend $z = c_1 \cdot c_2$ to $\mathbf{v} = \mathrm{ext}(c_1, c_2)$
- Permute $\mathbf{v}$: $T_{b_1,b_2}(\mathbf{v})$ for $b_1, b_2 \hookleftarrow \mathcal{U}(\{0,1\})$

Solution to the Sub-Problem

Goal

Prove that a secret bit z is of form $z = c_1 \cdot c_2$, while remaining able to prove that the c_1 and c_2 satisfy other equations.

$$\mathsf{v} = \mathrm{ext}(c_1, c_2) \iff T_{b_1,b_2}(\mathsf{v}) = \mathrm{ext}(c_1 \oplus b_1, c_2 \oplus b_2)$$

- Extend $z = c_1 \cdot c_2$ to $\mathbf{v} = \mathrm{ext}(c_1, c_2)$
- Permute $\mathbf{v}$: $T_{b_1,b_2}(\mathbf{v})$ for $b_1, b_2 \hookleftarrow \mathcal{U}(\{0,1\})$
- **same** bits c_1, c_2 appear in other equations $\Rightarrow$ **same** masks b_1, b_2

Group Encryption: Putting Everything Together

Ingredients

- Anonymous encryption — [ABB10] IBE + [CHK04] transform
- Signature scheme — [LLMNW16]
- Supporting ZK proofs — [Presented result]

+ Modular construction — [KTY07]

Group Encryption: Putting Everything Together

Ingredients

- ▶ Anonymous encryption — [ABB10] IBE + [CHK04] transform
- ▶ Signature scheme — [LLMNW16]
- ▶ Supporting ZK proofs — [Presented result]
- $+$ Modular construction — [KTY07]

- ▶ Our results (Libert-Ling-M-Nguyen-Wang, Asiacrypt'16):
  - ■ Zero-Knowledge arguments for "quadratic relations":

$$\boxed{\mathrm{X}} \cdot \boxed{\mathrm{s}} + \boxed{\mathrm{e}} = \boxed{\mathrm{b}} \bmod q.$$

$\rightarrow$ Building block for cryptography: may be of independent interest

  - ■ First construction of group encryption from (classical) lattice assumptions proven secure in the standard model

Outline

Practical group signature (AsiaCCS'16)

Pairings

First lattice-based signature with efficient protocols (Asiacrypt'16)

ZK argument of correct evaluation of committed branching programs (Asiacrypt'16)

Lattices

ZK argument for quadratic relations (Asiacrypt'17)

First Lattice-Based Signature with Efficient Protocols

(Libert-Ling-M-Nguyen-Wang, Asiacrypt'16)

A signature scheme $(\mathsf{Keygen}, \mathsf{Sign}_{\mathsf{sk}}, \mathsf{Verif}_{\mathsf{vk}})$ with efficient protocols[1]:

- To **sign** a committed value;
- To **prove** possession of a signature.

[1] Camenisch-Lysyanskaya, SCN'02

First Lattice-Based Signature with Efficient Protocols

(Libert-Ling-M-Nguyen-Wang, Asiacrypt'16)

A signature scheme $(\mathsf{Keygen}, \mathsf{Sign}_{\mathsf{sk}}, \mathsf{Verif}_{\mathsf{vk}})$ with efficient protocols[1]:

- To **sign** a committed value;
- To **prove** possession of a signature.

Security

- Unforgeability;
- Security of the two protocols;
- Anonymity.

$\rightarrow$ Many applications for privacy-based protocols.

[1] Camenisch-Lysyanskaya, SCN'02

First Lattice-Based Signature with Efficient Protocols

(Libert-Ling-M-Nguyen-Wang, Asiacrypt'16)

A signature scheme ($\mathsf{Keygen}, \mathsf{Sign}_{\mathsf{sk}}, \mathsf{Verif}_{\mathsf{vk}}$) with efficient protocols[1]:

- To **sign** a committed value;
- To **prove** possession of a signature.

Security

- Unforgeability;
- Security of the two protocols;
- Anonymity.

→ Many applications for privacy-based protocols.

✗ *Existing constructions rely on Strong RSA assumption or bilinear maps.*

[1] Camenisch-Lysyanskaya, SCN'02

Outline

Practical group signature (AsiaCCS'16)

Pairings

First lattice-based signature with efficient protocols (Asiacrypt'16)

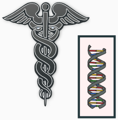
ZK argument of correct evaluation of committed branching programs (Asiacrypt'16)

Lattices

ZK argument for quadratic relations (Asiacrypt'17)

Adaptive Oblivious Transfer

(Naor-Pinkas, Crypto'99; Libert-Ling-M-Nguyen-Wang, Asiacrypt'17)

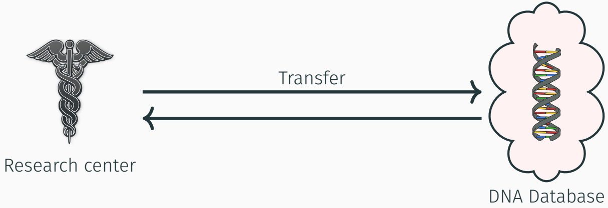


Research center

- DNA storage is expensive

Adaptive Oblivious Transfer

(Naor-Pinkas, Crypto'99; Libert-Ling-M-Nguyen-Wang, Asiacrypt'17)



- DNA storage is expensive
- DNA Database queries are sensitive

Adaptive Oblivious Transfer

(Naor-Pinkas, Crypto'99; Libert-Ling-M-Nguyen-Wang, Asiacrypt'17)

- DNA storage is expensive
- DNA Database queries are sensitive

⟶ make queries anonymous and unlinkable

Adaptive Oblivious Transfer

(Naor-Pinkas, Crypto'99; Libert-Ling-M-Nguyen-Wang, Asiacrypt'17)

- DNA storage is expensive
- DNA Database queries are sensitive

$\longrightarrow$ make queries anonymous and unlinkable

Extending expressiveness of Stern-like protocols
$\Rightarrow$ First construction from lattices with access control

Outline

Practical group signature (AsiaCCS'16)

Pairings

First lattice-based signature with efficient protocols (Asiacrypt'16)

ZK argument of correct evaluation of committed branching programs (Asiacrypt'16)

Lattices

ZK argument for quadratic relations (Asiacrypt'17)

Practical Group Signatures from Pairing Assumptions

(Libert-M-Peters-Yung, AsiaCCS'16)

Pairing

Let $\mathbb{G}, \hat{\mathbb{G}}$ and $\mathbb{G}_T$ be cyclic groups of prime order p.

$$e : \mathbb{G} \times \hat{\mathbb{G}} \longrightarrow \mathbb{G}_T$$

$$\forall g \in \mathbb{G}, \hat{g} \in \hat{\mathbb{G}}, a, b \in \mathbb{Z}, e(g^a, \hat{g}^b) = e(g, \hat{g})^{ab}$$

Hardness relies on (variant of) *Decision Diffie-Hellman*

— Pairings are not quantum-resistant (Shor 1999)

Practical Group Signatures from Pairing Assumptions

(Libert-M-Peters-Yung, AsiaCCS'16)

Pairing

Let $\mathbb{G}, \hat{\mathbb{G}}$ and $\mathbb{G}_T$ be cyclic groups of prime order p.

$$e : \mathbb{G} \times \hat{\mathbb{G}} \longrightarrow \mathbb{G}_T$$

$$\forall g \in \mathbb{G}, \hat{g} \in \hat{\mathbb{G}}, a, b \in \mathbb{Z}, e(g^a, \hat{g}^b) = e(g, \hat{g})^{ab}$$

Hardness relies on (variant of) *Decision Diffie-Hellman*

- – Pairings are not quantum-resistant (Shor 1999)
- \+ Pairings are more practical than lattices

Practical Group Signatures from Pairing Assumptions

(Libert-M-Peters-Yung, AsiaCCS'16)

Pairing

Let $\mathbb{G}, \hat{\mathbb{G}}$ and $\mathbb{G}_T$ be cyclic groups of prime order p.

$$e : \mathbb{G} \times \hat{\mathbb{G}} \longrightarrow \mathbb{G}_T$$

$$\forall g \in \mathbb{G}, \hat{g} \in \hat{\mathbb{G}}, a, b \in \mathbb{Z}, e(g^a, \hat{g}^b) = e(g, \hat{g})^{ab}$$

Hardness relies on (variant of) *Decision Diffie-Hellman*

– Pairings are not quantum-resistant (Shor 1999)

+ Pairings are more practical than lattices

- Design supported by an open-source implementation in C
- Use Relic toolkit

  `https://gforge.inria.fr/projects/sigmasig-c/`

Conclusion

My Research so far

Around two axes:

- (Privacy-preserving) protocol design
  - From pairings
  - From lattices
- Security proofs

Conclusion

My Research so far

Around two axes:

- (Privacy-preserving) protocol design
  - From pairings
  - From lattices
- Security proofs

Some disadvantages:

Adaptive OT Use of “LWE noise flooding”

Conclusion

My Research so far

Around two axes:

- (Privacy-preserving) protocol design
  - From pairings
  - From lattices
- Security proofs

Some disadvantages:

Adaptive OT Use of "LWE noise flooding"

Dynamic GS Use of lattice trapdoors

Conclusion

My Research so far

Around two axes:

- (Privacy-preserving) protocol design
  - From pairings
  - From lattices
- Security proofs

Some disadvantages:

Adaptive OT Use of “LWE noise flooding”

Dynamic GS Use of lattice trapdoors

Stern-like proofs Constant soundness error of 2/3

Open Problems

Follow-ups

- Universally composable oblivious transfer from LWE?
- More efficient compact e-cash system?

Zero-knowledge proofs

- Negligible soundness error for expressive statements in lattices?
- NIZK for NP from LWE?

Cryptographic constructions

- More efficient signatures (compatible with ZK proofs)?
- Efficient trapdoor-free (H)IBE?

Thank you for your Attention

Practical group signature (AsiaCCS'16)

Pairings

First lattice-based signature with efficient protocols (Asiacrypt'16)

ZK argument of correct evaluation of committed branching programs (Asiacrypt'16)

Lattices

ZK argument for quadratic relations (Asiacrypt'17)

What next? More protocol designs, zero-knowledge proofs and foundations of cryptographic constructions!