Simpler Efficient Group Signatures from Lattices

Phong Nguyen[1], Jiang Zhang[2], Zhenfeng Zhang[2]

[1]INRIA, France and Tsinghua University, China
[2]Institute of Software, Chinese Academy of Sciences

PKC 2015 (March 30 — April 1, 2015)
NIST — Gaithersburg, Maryland USA

Outline

TCA

Introduction
Group Signature

Digital signatures have been widely used to

ensure authenticity of {
- the signer 
- the document 

Introduction
Group Signature

Digital signatures have been widely used to

ensure authenticity of
- the signer 
- the document 

However, two privacy limitations:

1) the signer's identity is revealed;

2) multiple signatures are linkable.

Introduction
Group Signature

Group Signature is introduced by Chaum and van Heyst [CvH'91].
(static groups) $\Pi_{\text{GS}} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Open})$

User

σ

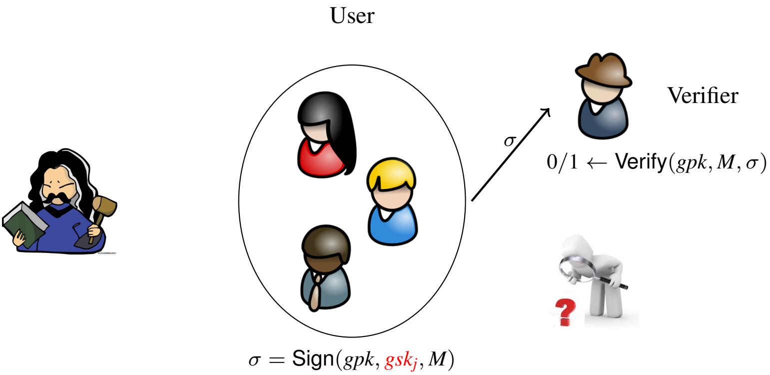
Verifier

$0/1 \leftarrow \mathsf{Verify}(gpk, M, \sigma)$

$\sigma = \mathsf{Sign}(gpk, gsk_j, M)$

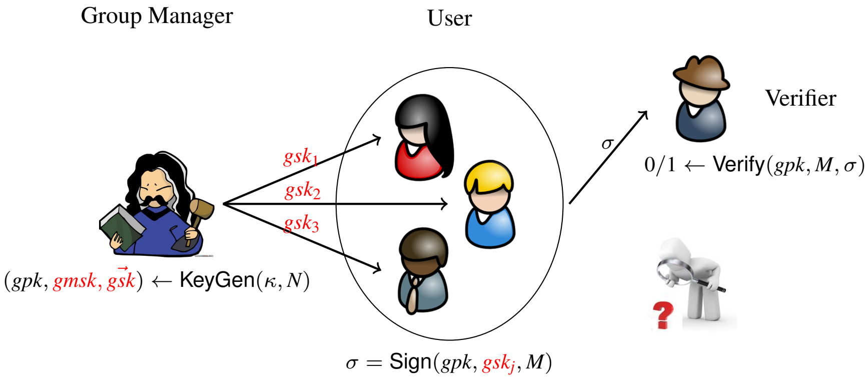
Introduction
Group Signature

Group Signature is introduced by Chaum and van Heyst [CvH'91].
(static groups) $\Pi_{\mathrm{GS}} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Open})$



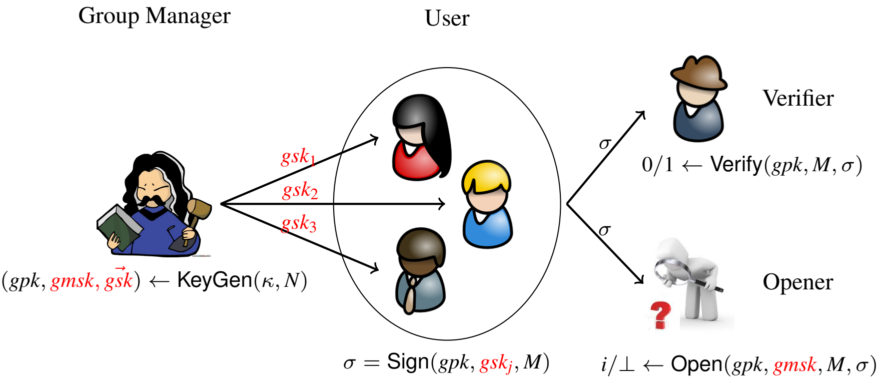
Introduction
Group Signature

Group Signature is introduced by Chaum and van Heyst [CvH'91].
(static groups) $\Pi_{\text{GS}} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Open})$



Introduction
Group Signature

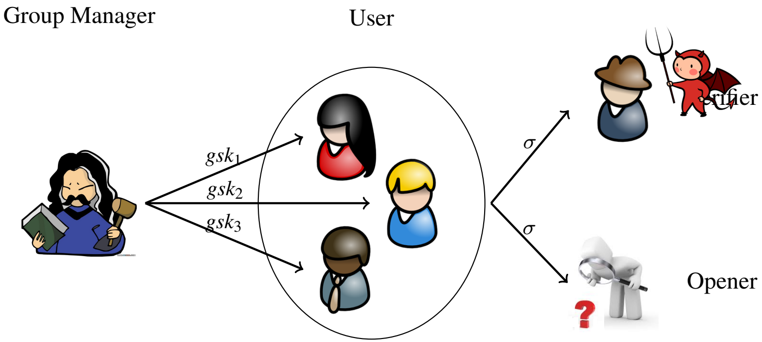
Group Signature is introduced by Chaum and van Heyst [CvH'91].
(static groups) $\Pi_{\text{GS}} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Open})$



Introduction

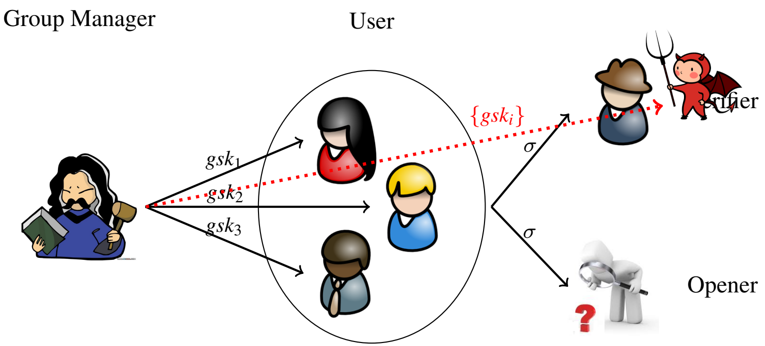
Group Signature

The security of a group signature: **full anonymity** & full traceability [BMW'03]

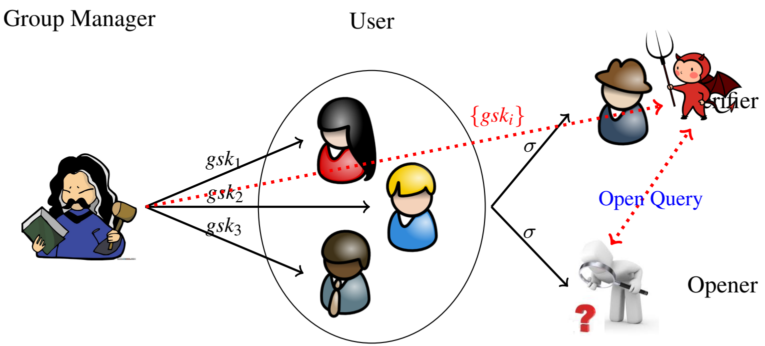


Introduction
Group Signature

The security of a group signature: **full anonymity** & full traceability [BMW'03]

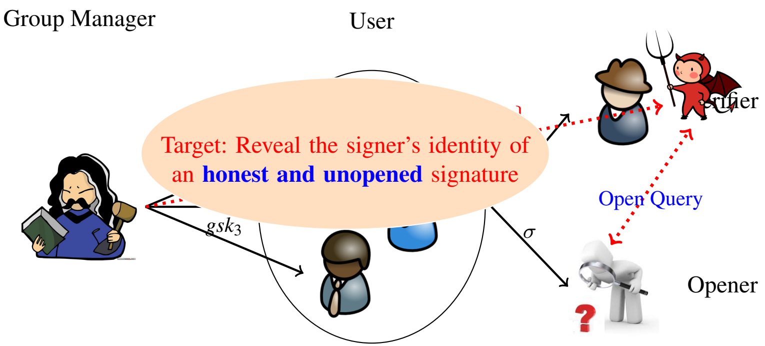


The security of a group signature: **full anonymity** & full traceability [BMW'03]



Introduction
Group Signature

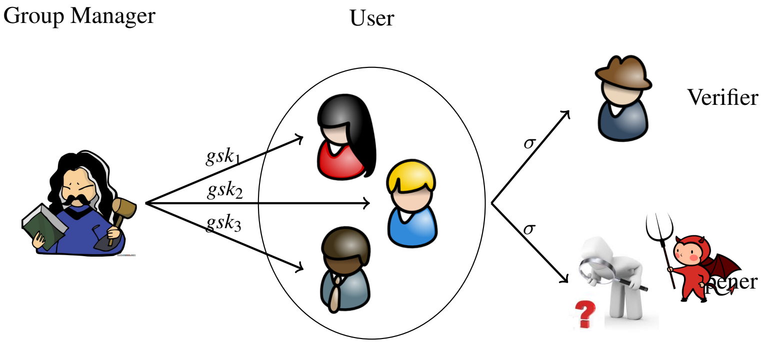
The security of a group signature: **full anonymity** & full traceability [BMW'03]



Introduction

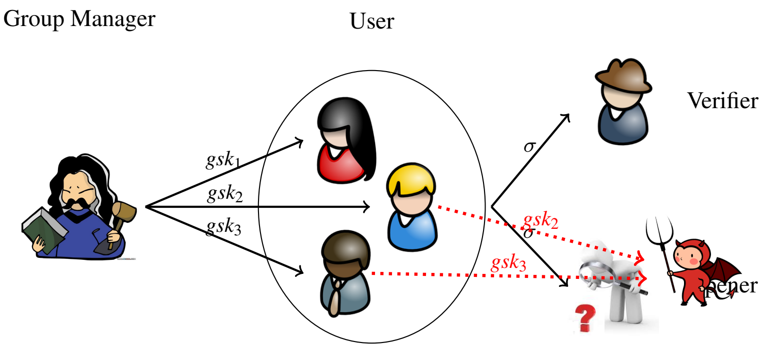
Group Signature

The security of a group signature: (fully) anonymity & **fully traceability** [BMW'03]



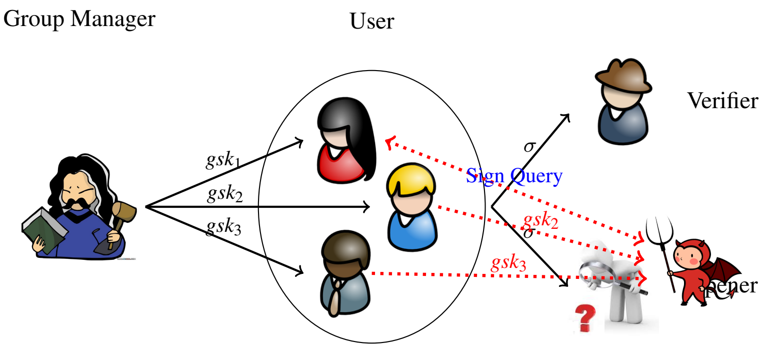
Introduction
Group Signature

The security of a group signature: (fully) anonymity & **fully traceability** [BMW'03]



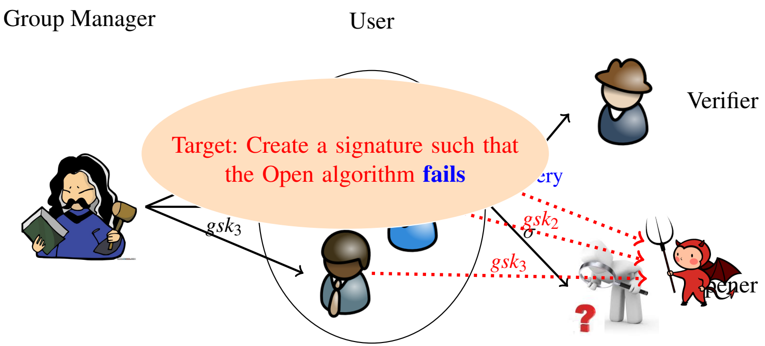
Introduction
Group Signature

The security of a group signature: (fully) anonymity & **fully traceability** [BMW'03]



Introduction
Group Signature

The security of a group signature: (fully) anonymity & **fully traceability** [BMW'03]



Introduction
The State of The Art

- Introduction of group signature [CvH'91]

$$\vdots$$

Introduction

The State of The Art

- Introduction of group signature [CvH'91]

$$\vdots$$

- Full anonymity and full traceability, BMW paradigm [BMW'03]

Introduction
The State of The Art

- Introduction of group signature [CvH'91]

$\vdots$

- Full anonymity and full traceability, BMW paradigm [BMW'03]
- CPA-anonymity, short and efficient construction [BBS'04]

Introduction
The State of The Art

- Introduction of group signature [CvH'91]

$\vdots$

- Full anonymity and full traceability, BMW paradigm [BMW'03]
- CPA-anonymity, short and efficient construction [BBS'04]
- "Constant size", "dynamic join", "membership revocation", [ACJT'00,CL'04,BW'06,BW'07,Groth'06,Groth'07,AFGHO'10,LPY'12]. . .

$\vdots$

Introduction

The State of The Art

- Introduction of group signature [CvH'91]

$\vdots$

- Full anonymity and full traceability, BMW paradigm [BMW'03]
- CPA-anonymity, short and efficient construction [BBS'04]
- "Constant size", "dynamic join", "membership revocation", [ACJT'00,CL'04,BW'06,BW'07,Groth'06,Groth'07,AFGHO'10,LPY'12]...

$\vdots$

Most of them are based on classic assumptions, e.g., strong RSA, sDH, DLIN, LRSW, $\cdots$

Introduction
The State of The Art

Lattice-based constructions ($N = \#(users)$):

- Gordon, Katz and Vaikuntanathan, ASIACRYPT 2010:
  $|gpk| = O(N)$, $|\sigma| = O(N)$

TCA

Introduction

The State of The Art

Lattice-based constructions ($N = \#(users)$):

- Gordon, Katz and Vaikuntanathan, ASIACRYPT 2010:
  $|gpk| = O(N)$, $|\sigma| = O(N)$
- Laguillaumie *et al.* [LLLS'13], ASIACRYPT 2013: Logarithmic efficiency,
  $|gpk| = O(\log N)$, $|\sigma| = O(\log N)$

Lattice-based constructions ($N = \#(users)$):

- Gordon, Katz and Vaikuntanathan, ASIACRYPT 2010:
  $|gpk| = O(N)$, $|\sigma| = O(N)$
- Laguillaumie *et al.* [LLLS'13], ASIACRYPT 2013: Logarithmic efficiency,
  $|gpk| = O(\log N)$, $|\sigma| = O(\log N)$
- Langlois *et al.* [LLNW'14], PKC 2014: Verifier local revocation,
  $|gpk| = O(\log N)$, $|\sigma| = O(\log N)$

Introduction
The State of The Art

Lattice-based constructions ($N = \#(users)$):

- Gordon, Katz and Vaikuntanathan, ASIACRYPT 2010: $|gpk| = O(N)$, $|\sigma| = O(N)$
- Laguillaumie *et al.* [LLLS'13], ASIACRYPT 2013: Logarithmic efficiency, $|gpk| = O(\log N)$, $|\sigma| = O(\log N)$
- Langlois *et al.* [LLNW'14], PKC 2014: Verifier local revocation, $|gpk| = O(\log N)$, $|\sigma| = O(\log N)$
- Very recently, Ling, Nguyen and Wang, PKC 2015: Tighter reduction, $|gpk| = O(\log N)$, $|\sigma| = O(\log N)$

Introduction

The State of The Art

Lattice-based constructions ($N = \#(users)$):

- Gordon, Katz and Vaikuntanathan, ASIACRYPT 2010:
  $|gpk| = O(N)$, $|\sigma| = O(N)$
- Laguillaumie *et al.* [LLLS'13], ASIACRYPT 2013: Logarithmic efficiency,
  $|gpk| = O(\log N)$, $|\sigma| = O(\log N)$
- Langlois *et al.* [LLNW'14], PKC 2014: Verifier local revocation,
  $|gpk| = O(\log N)$, $|\sigma| = O(\log N)$
- Very recently, Ling, Nguyen and Wang, PKC 2015: Tighter reduction,
  $|gpk| = O(\log N)$, $|\sigma| = O(\log N)$

The BMW paradigm

Introduction

The State of The Art

Lattice-based constructions ($N = \#(users)$):

- Gordon, Katz and Vaikuntanathan, ASIACRYPT 2010:
  $|gpk| = O(N)$, $|\sigma| = O(N)$
- Laguillaumie *et al.* [LLLS'13], ASIACRYPT 2013: Logarithmic efficiency,
  $|gpk| = O(\log N)$, $|\sigma| = O(\log N)$
- Langlois *et al.* [LLNW'14], PKC 2014: Verifier local revocation,
  $|gpk| = O(\log N)$, $|\sigma| = O(\log N)$
- Very recently, Ling, Nguyen and Wang, PKC 2015: Tighter reduction,
  $|gpk| = O(\log N)$, $|\sigma| = O(\log N)$

The BMW paradigm

LWE + SIS

We give a simpler and efficient construction,
almost reducing both $|gpk|$ and $|\sigma|$
by a factor of $O(\log N)$

We give a simpler and efficient construction,
almost reducing both $|gpk|$ and $|\sigma|$
by a factor of $O(\log N)$

Our scheme takes advantage of

an efficient encoding and a **new NIZK**

Introduction
This Work

We give a simpler and efficient construction,
almost reducing both $|gpk|$ and $|\sigma|$
by a factor of $O(\log N)$

Our scheme takes advantage of

an efficient encoding and a **new NIZK**

We introduce a new problem–**Split-SIS** ($\overset{c}{\approx}$ the standard SIS)

Security: **LWE + Split-SIS**

Our Approach
Lattices and Hard Problems

Let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, define m-dimensional full-rank integer lattice:

$$\Lambda_q^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m \;\; s.t. \; \mathbf{A}\mathbf{x} = 0 \mod q\}$$

Our Approach
Lattices and Hard Problems

Let $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, define m-dimensional full-rank integer lattice:

$$\Lambda_q^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m \;\; s.t. \; \mathbf{A}\mathbf{x} = 0 \mod q\}$$

Useful Facts:

- Generate a "uniform" $\mathbf{A}$ with a "trapdoor" [Ajtai'96,Peikert'09,MP'12]

Our Approach
Lattices and Hard Problems

Let $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, define m-dimensional full-rank integer lattice:

$$\Lambda_q^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m \;\; s.t.\; \mathbf{A}\mathbf{x} = 0 \mod q\}$$

Useful Facts:

- Generate a "uniform" **A** with a "trapdoor" [Ajtai'96,Peikert'09,MP'12]
- Sample "short vectors" from $\Lambda_q^{\perp}(\mathbf{A})$ [GPV'08,AP'09,MP'12]

Our Approach

Lattices and Hard Problems

$$m\left\{\underbrace{\mathbf{A}^T}_{n}\right. \times \mathbf{s} + \mathbf{e} = \mathbf{b} \mod q$$

Fixed $\mathbf{s} \in \mathbb{Z}_q^n$, and "noise" χ, define

$$A_{\mathbf{s},\chi} = \{(\mathbf{u}, \mathbf{u}^T\mathbf{s} + e) \mid \mathbf{u} \leftarrow_r \mathbb{Z}_q^n, e \leftarrow_r \chi\}$$

Learning with errors (LWE):

- Computational LWE: Given polynomial samples, find $\mathbf{s}$

Our Approach
Lattices and Hard Problems

$$m\left\{ \underbrace{\boxed{\mathbf{A}^T}}_{n} \right. \times \boxed{\mathbf{s}} + \boxed{\mathbf{e}} = \boxed{\mathbf{b}} \mod q$$

Fixed $\mathbf{s} \in \mathbb{Z}_q^n$, and "noise" χ, define

$$A_{\mathbf{s},\chi} = \{(\mathbf{u}, \mathbf{u}^T\mathbf{s} + e) \mid \mathbf{u} \leftarrow_r \mathbb{Z}_q^n, e \leftarrow_r \chi\}$$

Learning with errors (LWE):

- Computational LWE: Given polynomial samples, find $\mathbf{s}$
- Decisional LWE: Distinguish $A_{\mathbf{s},\chi}$ from $\mathcal{U}(\mathbb{Z}_q^n \times \mathbb{Z}_q)$

Our Approach

Lattices and Hard Problems

$$n\left\{\boxed{\mathbf{A}}\right. \times \boxed{\mathbf{s}} = \boxed{\mathbf{u}} \mod q$$

$$\underbrace{\boxed{\mathbf{A}}}_{m}$$

Small Integer Solution (SIS):

Given $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, find "small" $\mathbf{s} \in \mathbb{Z}_q^m \backslash \{\mathbf{0}\}$, s.t., $\mathbf{A}\mathbf{s} = \mathbf{0} \bmod q$

Inhomogeneous Small Integer Solution (ISIS):

Given $(\mathbf{A}, \mathbf{u}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^n$, find "small" $\mathbf{s} \in \mathbb{Z}_q^m$, s.t., $\mathbf{A}\mathbf{s} = \mathbf{u} \bmod q$

Our Approach
Lattices and Hard Problems

$$n\left\{\underbrace{\boxed{\mathbf{A}}}_{m}\right. \times \boxed{\mathbf{s}} = \boxed{\mathbf{u}} \mod q$$

Small Integer Solution (SIS):

Given $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, find "small" $\mathbf{s} \in \mathbb{Z}_q^m \backslash \{\mathbf{0}\}$, s.t., $\mathbf{As} = \mathbf{0} \bmod q$

Inhomogeneous Small Integer Solution (ISIS):

Given $(\mathbf{A}, \mathbf{u}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^n$, find "small" $\mathbf{s} \in \mathbb{Z}_q^m$, s.t., $\mathbf{As} = \mathbf{u} \bmod q$

Both LWE and SIS (ISIS) $\overset{c}{\approx}$ SIVP$_\gamma$ in the worst case [Ajtai'96,Regev'05,. . .]

Our Approach
The BMW Paradigm

We first recall the BMW paradigm:

- KeyGen(κ, N):
  1. Generate the group public key gpk;
  2. Find an "identity encoding" $H(gpk, j)$;
  3. derive user secret key gsk_j corresponding to $H(gpk, j)$.

Our Approach
The BMW Paradigm

We first recall the BMW paradigm:

- $\mathsf{KeyGen}(\kappa, N)$:
  1. Generate the group public key gpk;
  2. Find an "identity encoding" $H(gpk, j)$;
  3. derive user secret key gsk_j corresponding to $H(gpk, j)$.

- $\mathsf{Sign}(gpk, gsk_j, M)$:
  1. Generate a proof π that gsk_j satisfies the relation determined by $H(gpk, j)$
  2. Return $\sigma = \pi$

Our Approach
The BMW Paradigm

We first recall the BMW paradigm:

- KeyGen(κ, N):
  1. Generate the group public key gpk;
  2. Find an "identity encoding" $H(gpk, j)$;
  3. derive user secret key gsk_j corresponding to $H(gpk, j)$.

- Sign(gpk, gsk_j, M):
  1. Generate a proof π that gsk_j satisfies the relation determined by $H(gpk, j)$
  2. Return $\sigma = \pi$

Key Issue: Find an encoding $H(gpk, j)$ and an NIZK for $H(gpk, j)$!

TCA

Our Approach
The BMW Paradigm

Both constructions [GKV'10,LLLS'13] follow the BMW paradigm:

- Gordon, Katz and Vaikuntanathan, ASIACRYPT 2010:

  $$gpk = (\mathbf{A}_1, \ldots, \mathbf{A}_N),$$
  $$H(gpk, j) = \mathbf{A}_j$$

  Both $|gpk|$ and $|\sigma|$ have linear size

Our Approach
The BMW Paradigm

Both constructions [GKV'10,LLLS'13] follow the BMW paradigm:

- Gordon, Katz and Vaikuntanathan, ASIACRYPT 2010:
  $gpk = (\mathbf{A}_1, \ldots, \mathbf{A}_N)$,
  $H(gpk, j) = \mathbf{A}_j$

  Both $|gpk|$ and $|\sigma|$ have linear size

- Laguillaumie *et al.* [LLLS'13], ASIACRYPT 2013:
  $gpk = (\mathbf{A}_1, \ldots, \mathbf{A}_\ell)$, where $\ell = \log N$,
  $H(gpk, j) = \sum_{i=1}^{i=\ell} j_i \mathbf{A}_j$, where $(j_1, \ldots, j_\ell)$—binary decomposition of j

  Both $|gpk|$ and $|\sigma|$ have logarithmic size

Our Approach

Our Initial Attempt

How about the efficient encoding function used in IBE [ABB'10]?

Full rank difference $G : \mathbb{Z}_q \to \mathbb{Z}_q^{n\times n}$

$$gpk = (\mathbf{A}_1, \mathbf{A}_{2,1}, \mathbf{A}_{2,2}),$$
$$H(gpk, j) = (\mathbf{A}_1 \| \mathbf{A}_{2,1} + G(j)\mathbf{A}_{2,2})$$

Our Approach
Our Initial Attempt

How about the efficient encoding function used in IBE [ABB'10]?

Full rank difference $G : \mathbb{Z}_q \to \mathbb{Z}_q^{n\times n}$

$$gpk = (\mathbf{A}_1, \mathbf{A}_{2,1}, \mathbf{A}_{2,2}),$$
$$H(gpk, j) = (\mathbf{A}_1 \| \mathbf{A}_{2,1} + G(j)\mathbf{A}_{2,2})$$

- KeyGen(κ, N):
  1. Generate $gpk = (\mathbf{A}_1, \mathbf{A}_{2,1}, \mathbf{A}_{2,2})$ with a trapdoor of $\mathbf{A}_1$;
  2. Define $\mathbf{A}_j := H(gpk, j) = (\mathbf{A}_1 \| \mathbf{A}_{2,1} + G(j)\mathbf{A}_{2,2})$;
  3. Sample a short vector $gsk_j = \mathbf{x}_j = (\mathbf{x}_{j,1}, \mathbf{x}_{j,2})$ from $\Lambda_q^{\perp}(\mathbf{A}_j)$.

Our Approach
Our Initial Attempt

How about the efficient encoding function used in IBE [ABB'10]?

Full rank difference $G : \mathbb{Z}_q \to \mathbb{Z}_q^{n\times n}$

$$gpk = (\mathbf{A}_1, \mathbf{A}_{2,1}, \mathbf{A}_{2,2}),$$
$$H(gpk, j) = (\mathbf{A}_1 \| \mathbf{A}_{2,1} + G(j)\mathbf{A}_{2,2})$$

- KeyGen(κ, N):
  1. Generate $gpk = (\mathbf{A}_1, \mathbf{A}_{2,1}, \mathbf{A}_{2,2})$ with a trapdoor of $\mathbf{A}_1$;
  2. Define $\mathbf{A}_j := H(gpk, j) = (\mathbf{A}_1 \| \mathbf{A}_{2,1} + G(j)\mathbf{A}_{2,2})$;
  3. Sample a short vector $gsk_j = \mathbf{x}_j = (\mathbf{x}_{j,1}, \mathbf{x}_{j,2})$ from $\Lambda_q^{\perp}(\mathbf{A}_j)$.
- Sign(gpk, gsk_j, M):
  1. Generate a proof π that $gsk_j = (\mathbf{x}_{j,1}, \mathbf{x}_{j,2})$ and j satisfy
     1) gsk_j is short, and
     2) $\mathbf{A}_1\mathbf{x}_{j,1} + (\mathbf{A}_{2,1} + G(j)\mathbf{A}_{2,2})\mathbf{x}_{j,2} = \mathbf{0}$
  2. Return $\sigma = \pi$

Our Approach
Our Initial Attempt

How about the efficient encoding function used in IBE [ABB'10]?

Full rank difference $G : \mathbb{Z}_q \to \mathbb{Z}_q^{n\times n}$

$$gpk = (\mathbf{A}_1, \mathbf{A}_{2,1}, \mathbf{A}_{2,2}),$$
$$H(gpk, j) = (\mathbf{A}_1 \| \mathbf{A}_{2,1} + G(j)\mathbf{A}_{2,2})$$

- KeyGen(κ, N):
  1. Generate $gpk = (\mathbf{A}_1, \mathbf{A}_{2,1}, \mathbf{A}_{2,2})$ with a trapdoor of $\mathbf{A}_1$;
  2. Define $\mathbf{A}_j := H(gpk, j) = (\mathbf{A}_1 \| \mathbf{A}_{2,1} + G(j)\mathbf{A}_{2,2})$;
  3. Sample a short vector $gsk_j = \mathbf{x}_j = (\mathbf{x}_{j,1}, \mathbf{x}_{j,2})$ from $\Lambda_q^{\perp}(\mathbf{A}_j)$.
- Sign(gpk, gsk_j, M):
  1. Generate a proof π that $gsk_j = (\mathbf{x}_{j,1}, \mathbf{x}_{j,2})$ and j satisfy
     1) gsk_j is short, and
     2) $\mathbf{A}_1\mathbf{x}_{j,1} + (\mathbf{A}_{2,1} + G(j)\mathbf{A}_{2,2})\mathbf{x}_{j,2} = \mathbf{0}$
  2. Return $\sigma = \pi$

But we cannot efficiently prove $\mathbf{A}_1\mathbf{x}_{j,1} + (\mathbf{A}_{2,1} + G(j)\mathbf{A}_{2,2})\mathbf{x}_{j,2} = \mathbf{0}$

Our Approach
Our Initial Attempt

Instead, we use a simple identity function $G(j) = j$

- KeyGen(κ, N):
  1. Generate $gpk = (\mathbf{A}_1, \mathbf{A}_{2,1}, \mathbf{A}_{2,2})$ with a trapdoor of $\mathbf{A}_1$;
  2. Define $\mathbf{A}_j := H(gpk, j) = (\mathbf{A}_1 \| \mathbf{A}_{2,1} + G(j)\mathbf{A}_{2,2})$;
  3. Sample a short vector $gsk_j = \mathbf{x}_j = (\mathbf{x}_{j,1}, \mathbf{x}_{j,2})$ from $\Lambda_q^\perp(\mathbf{A}_j)$.
- Sign(gpk, gsk_j, M):
  1. Generate a proof π that $gsk_j = (\mathbf{x}_{j,1}, \mathbf{x}_{j,2})$ and j satisfy
     1) gsk_j is short, and
     2) $\mathbf{A}_1\mathbf{x}_{j,1} + (\mathbf{A}_{2,1} + j\mathbf{A}_{2,2})\mathbf{x}_{j,2} = \mathbf{0}$
  2. Return $\sigma = \pi$

Our Approach

Our Initial Attempt

Instead, we use a simple identity function $G(j) = j$

- KeyGen(κ, N):
  1. Generate $gpk = (\mathbf{A}_1, \mathbf{A}_{2,1}, \mathbf{A}_{2,2})$ with a trapdoor of $\mathbf{A}_1$;
  2. Define $\mathbf{A}_j := H(gpk, j) = (\mathbf{A}_1 \| \mathbf{A}_{2,1} + G(j)\mathbf{A}_{2,2})$;
  3. Sample a short vector $gsk_j = \mathbf{x}_j = (\mathbf{x}_{j,1}, \mathbf{x}_{j,2})$ from $\Lambda_q^{\perp}(\mathbf{A}_j)$.
- Sign(gpk, gsk_j, M):
  1. Generate a proof π that $gsk_j = (\mathbf{x}_{j,1}, \mathbf{x}_{j,2})$ and j satisfy

     1) gsk_j is short, and

     2) $\mathbf{A}_1\mathbf{x}_{j,1} + (\mathbf{A}_{2,1} + j\mathbf{A}_{2,2})\mathbf{x}_{j,2} = \mathbf{0}$
  2. Return $\sigma = \pi$

Let $\mathbf{b} = \mathbf{A}_{2,2}\mathbf{x}_{j,2}$, we have

$$\mathbf{A}_1\mathbf{x}_{j,1} + j\mathbf{b} = (\mathbf{A}_1 \| \mathbf{b})(\mathbf{x}_{j,1}; j) = -\mathbf{A}_{2,1}\mathbf{x}_{j,2}$$

A variant of ISIS

The Split-SIS Problem
The Description

Given $\mathbf{A} = (\mathbf{A}_1, \mathbf{A}_2) \in \mathbb{Z}_q^{n \times (m_1+m_2)}$,

Small Integer Solution (SIS): find "small" $\mathbf{x} \in \mathbb{Z}_q^{m_1+m_2}/\{\mathbf{0}\}$, s.t., $\mathbf{Ax} = \mathbf{0} \bmod q$.

The Split-SIS Problem
The Description

Given $\mathbf{A} = (\mathbf{A}_1, \mathbf{A}_2) \in \mathbb{Z}_q^{n\times(m_1+m_2)}$,

Small Integer Solution (SIS): find "small" $\mathbf{x} \in \mathbb{Z}_q^{m_1+m_2}/\{\mathbf{0}\}$, s.t., $\mathbf{A}\mathbf{x} = \mathbf{0} \bmod q$.

Split-SIS: find $h \in \mathbb{Z}_q$ and 'small" $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2) \in \mathbb{Z}_q^{m_1+m_2}/\{\mathbf{0}\}$, s.t.,

$$\mathbf{A}_1\mathbf{x}_1 + h\mathbf{A}_2\mathbf{x}_2 = \mathbf{0} \bmod q \qquad \wedge \qquad (\mathbf{x}_1; h\mathbf{x}_2) \neq \mathbf{0}$$

The Split-SIS Problem

The Description

Given $\mathbf{A} = (\mathbf{A}_1, \mathbf{A}_2) \in \mathbb{Z}_q^{n\times(m_1+m_2)}$,

Small Integer Solution (SIS): find "small" $\mathbf{x} \in \mathbb{Z}_q^{m_1+m_2}/\{\mathbf{0}\}$, s.t., $\mathbf{Ax} = \mathbf{0} \bmod q$.

Split-SIS: find $h \in \mathbb{Z}_q$ and 'small" $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2) \in \mathbb{Z}_q^{m_1+m_2}/\{\mathbf{0}\}$, s.t.,

$$\mathbf{A}_1\mathbf{x}_1 + h\mathbf{A}_2\mathbf{x}_2 = \mathbf{0} \bmod q \qquad \wedge \qquad (\mathbf{x}_1; h\mathbf{x}_2) \neq \mathbf{0}$$

For appropriate parameters, we prove that

Split-SIS is as hard as the standard SIS problem!

The Split-SIS Problem

A Hash Family from Split-SIS

Define a family of functions $\mathcal{H}$ with index $\mathbf{A}_1, \mathbf{A}_{2,2} \in \mathbb{Z}_q^{n \times m}$:

$$f_{\mathbf{A}_1, \mathbf{A}_{2,2}}(\mathbf{x}_1, \mathbf{x}_2, h) = (\mathbf{A}_1 \mathbf{x}_1 + h\mathbf{A}_{2,2}\mathbf{x}_2 \bmod q, \mathbf{x}_2)$$

The Split-SIS Problem

A Hash Family from Split-SIS

Define a family of functions $\mathcal{H}$ with index $\mathbf{A}_1, \mathbf{A}_{2,2} \in \mathbb{Z}_q^{n\times m}$:

$$f_{\mathbf{A}_1,\mathbf{A}_{2,2}}(\mathbf{x}_1, \mathbf{x}_2, h) = (\mathbf{A}_1\mathbf{x}_1 + h\mathbf{A}_{2,2}\mathbf{x}_2 \bmod q, \mathbf{x}_2)$$

We directly output the second input $\mathbf{x}_2$

The Split-SIS Problem

A Hash Family from Split-SIS

Define a family of functions $\mathcal{H}$ with index $\mathbf{A}_1, \mathbf{A}_{2,2} \in \mathbb{Z}_q^{n \times m}$:

$$f_{\mathbf{A}_1,\mathbf{A}_{2,2}}(\mathbf{x}_1, \mathbf{x}_2, h) = (\mathbf{A}_1\mathbf{x}_1 + h\mathbf{A}_{2,2}\mathbf{x}_2 \bmod q, \mathbf{x}_2)$$

We directly output the second input $\mathbf{x}_2$

If Split-SIS is hard, then for some parameters $\mathcal{H}$ is

one-way, collision-resistant, and statistically hiding "h"

The Split-SIS Problem
A Hash Family from Split-SIS

Define a family of functions $\mathcal{H}$ with index $\mathbf{A}_1, \mathbf{A}_{2,2} \in \mathbb{Z}_q^{n\times m}$:

$$f_{\mathbf{A}_1,\mathbf{A}_{2,2}}(\mathbf{x}_1, \mathbf{x}_2, h) = (\mathbf{A}_1\mathbf{x}_1 + h\mathbf{A}_{2,2}\mathbf{x}_2 \bmod q, \mathbf{x}_2)$$

We directly output the second input $\mathbf{x}_2$

Given $(\mathbf{A}_1, \mathbf{A}_{2,2})$ and $\mathbf{y} = (\mathbf{y_1}, \mathbf{y_2})$, prove there exists $(\mathbf{x}_1, \mathbf{x}_2, h)$ such that

$$f_{\mathbf{A}_1,\mathbf{A}_{2,2}}(\mathbf{x}_1, \mathbf{x}_2, h) = \mathbf{y}$$
$$\Updownarrow$$
$$(\mathbf{A}_1\|\mathbf{b})(\mathbf{x}_1; h) = \mathbf{y}_1 \text{ for } \mathbf{b} = \mathbf{A}_{2,2}\mathbf{y}_2$$

The Split-SIS Problem
The Modified Construction

- KeyGen(κ, N):
  1. Generate $gpk = (\mathbf{A}_1, \mathbf{A}_{2,1}, \mathbf{A}_{2,2})$ with a trapdoor of $\mathbf{A}_1$;
  2. Define $\mathbf{A}_j := H(gpk, j) = (\mathbf{A}_1 \| \mathbf{A}_{2,1} + j\mathbf{A}_{2,2})$;
  3. Compute a trapdoor $gsk_j = \mathbf{T}_{\mathbf{A}_j}$ of $\mathbf{A}_j$.

The Split-SIS Problem

The Modified Construction

- KeyGen(κ, N):
  1. Generate $gpk = (\mathbf{A}_1, \mathbf{A}_{2,1}, \mathbf{A}_{2,2})$ with a trapdoor of $\mathbf{A}_1$;
  2. Define $\mathbf{A}_j := H(gpk, j) = (\mathbf{A}_1 \| \mathbf{A}_{2,1} + j\mathbf{A}_{2,2})$;
  3. Compute a trapdoor $gsk_j = \mathbf{T}_{\mathbf{A}_j}$ of $\mathbf{A}_j$.

- Sign(gpk, gsk_j, M):
  1. Use gsk_j to sample a short vector $\mathbf{x}_j = (\mathbf{x}_{j,1}, \mathbf{x}_{j,2})$ from $\Lambda_q^{\perp}(\mathbf{A}_j)$;
  2. Compute $\mathbf{b} = \mathbf{A}_{2,2}\mathbf{x}_{j,2}$ and $\mathbf{y} = -\mathbf{A}_{2,1}\mathbf{x}_{j,1}$;
  3. Generate a proof π that $\mathbf{x}_{j,1}$ and j satisfy $(\mathbf{A}_1 \| \mathbf{b})(\mathbf{x}_{j,1}; j) = \mathbf{y}$;
  4. Return $\sigma = (\mathbf{x}_{j,2}, \pi)$.

The Split-SIS Problem

The Modified Construction

- KeyGen(κ, N):
  1. Generate $gpk = (\mathbf{A}_1, \mathbf{A}_{2,1}, \mathbf{A}_{2,2})$ with a trapdoor of $\mathbf{A}_1$;
  2. Define $\mathbf{A}_j := H(gpk, j) = (\mathbf{A}_1 \| \mathbf{A}_{2,1} + j\mathbf{A}_{2,2})$;
  3. Compute a trapdoor $gsk_j = \mathbf{T}_{\mathbf{A}_j}$ of $\mathbf{A}_j$.

- Sign(gpk, gsk_j, M):
  1. Use gsk_j to sample a short vector $\mathbf{x}_j = (\mathbf{x}_{j,1}, \mathbf{x}_{j,2})$ from $\Lambda_q^\perp(\mathbf{A}_j)$;
  2. Compute $\mathbf{b} = \mathbf{A}_{2,2}\mathbf{x}_{j,2}$ and $\mathbf{y} = -\mathbf{A}_{2,1}\mathbf{x}_{j,1}$;
  3. Generate a proof π that $\mathbf{x}_{j,1}$ and j satisfy $(\mathbf{A}_1 \| \mathbf{b})(\mathbf{x}_{j,1}; j) = \mathbf{y}$;
  4. Return $\sigma = (\mathbf{x}_{j,2}, \pi)$.

$\mathbf{x}_{j,2}$ is statistically indistinguishable w.r.t. j

Conclusion

We give a simpler and efficient construction, almost reducing both $|gpk|$ and $|\sigma|$ by a factor of $O(\log N)$

Conclusion

We give a simpler and efficient construction,
almost reducing both $|gpk|$ and $|\sigma|$
by a factor of $O(\log N)$

We are so close to **"Constant Size"**


THANK
YOU!