Signature Schemes with Efficient Protocols and Dynamic Group Signatures from Lattice Assumptions

Benoît Libert\textsuperscript{1,2}  San Ling\textsuperscript{3}  Fabrice Mouhartem\textsuperscript{1}  
Khoa Nguyen\textsuperscript{3}  Huaxiong Wang\textsuperscript{3}

\textsuperscript{1}É.N.S. de Lyon, France  
\textsuperscript{2}CNRS, France  
\textsuperscript{3}Nanyang Technological University, Singapore

Caen, 30/11/2016
Privacy-Preserving Cryptography

**Important Goal:** Anonymous authentication.
Privacy-Preserving Cryptography

**Important Goal:** Anonymous authentication.

e.g., e-voting, e-cash, group signatures, anonymous credentials...
Privacy-Preserving Cryptography

**Important Goal:** Anonymous authentication.

e.g., e-voting, e-cash, group signatures, anonymous credentials.

Ingredients

- A signature scheme
- Zero-knowledge (ZK) proofs
Privacy-Preserving Cryptography

**Important Goal:** Anonymous authentication.

e.g., e-voting, e-cash, group signatures, anonymous credentials...

**Ingredients**

- A signature scheme
- Zero-knowledge (ZK) proofs compatible with this signature (no hash functions)
Privacy-Preserving Cryptography

**Important Goal:** Anonymous authentication.

e.g., e-voting, e-cash, group signatures, anonymous credentials...

Ingredients

- A signature scheme
- Zero-knowledge (ZK) proofs compatible with this signature (no hash functions)
Digital Signatures

Signature Schemes

Signer

 Verify

Message

Signature

Veriﬁer

Sign
Digital Signatures

Guarantees *authenticity* and *integrity*. 
Group Signatures

A user wants to take public transportations.
Group Signatures

A user wants to take public transportations.

timestamp
Group Signatures

A user wants to take public transportations.

- Authenticity & Integrity
Group Signatures

A user wants to take public transportations.

- Authenticity & Integrity
- Anonymity
Group Signatures

A user wants to take public transportations.

- Authenticity & Integrity
- Anonymity
- Dynamicity

Join

signature
Group Signatures

A user wants to take public transportations.

- Authenticity & Integrity
- Anonymity
- Dynamicity
- Traceability
Why dynamic group signature?

**Dynamic group signatures**

In *dynamic* group signatures, new group members can be introduced *at any time*.

**Applications:** access control in public transportation, smart cars communications, anonymous access control (e.g., in buildings)…
Why dynamic group signature?

**Dynamic group signatures**

In *dynamic* group signatures, new group members can be introduced *at any time*.

**Applications:** access control in public transportation, smart cars communications, anonymous access control (e.g., in buildings)...

**Main Differences**

<table>
<thead>
<tr>
<th>Static Group</th>
<th>Dynamic Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>GM distributes keys</td>
<td>$\mathcal{U}_i$ makes his secret certified</td>
</tr>
<tr>
<td>Cannot add new users</td>
<td>Even colluding GM/OA cannot sign on behalf of a honest group member</td>
</tr>
</tbody>
</table>
Motivation

Advantages of the *dynamic* group setting:

- Add users without re-running the *Setup* phase;
Motivation

Advantages of the dynamic group setting:

► Add users without re-running the Setup phase;

► Even if everyone, including authorities, is dishonest, no one can sign in your name;
Motivation

Advantages of the \textit{dynamic} group setting:

- Add users without re-running the \texttt{Setup} phase;
- Even if everyone, including authorities, is dishonest, no one can sign in your name;
- Most use cases inherently require dynamic groups (e.g., building’s access control)
Commitments

Digital equivalent of a sealed box.

e.g., Pedersen Commitment

\[ pk = (g, h) \leftarrow \mathbb{G}^2 \]

\[ com = g^m \cdot h^r \]

\[ open = (m, r) \]
Commitments

Digital equivalent of a sealed box.

e.g., Pedersen Commitment

\[ pk = (g, h) \leftarrow \mathbb{G}^2 \]

\[ com = g^m \cdot h^r \]

\[ open = (m, r) \]

Properties

Commitments provide

- **Binding** property: once sealed, a value cannot be changed
- **Hiding** property: nobody is able tell what is inside the box without the key
## Anonymous Credentials (Chaum'85, Camenisch-Lysyanskya'01)

### Principle (e.g., U-Prove, Idemix)

Involves three parties: **Issuers, Users and Verifiers.**

- User dynamically obtains credentials from an issuer under a (pseudonym = commitment to a digital identity)
- ...and can dynamically prove possession of credentials using different *(unlinkable)* pseudonyms

### Different flavors: one-show/multi-show credentials, attribute-based access control, ...
Anonymous Credentials (Chaum'85, Camenisch-Lysyanskya’01)

**Principle** (e.g., U-Prove, Idemix)

Involves three parties: **Issuers, Users and Verifiers**.

- User dynamically obtains credentials from an issuer under a (pseudonym = commitment to a digital identity)
- ...and can dynamically prove possession of credentials using different *(unlinkable)* pseudonyms

**Different flavors**: one-show/multi-show credentials, attribute-based access control,...

**General construction** from signature with efficient protocols:

- Issuer gives a user a signature on a committed message;
- User proves that same secret underlies different pseudonyms;
- User proves that he possesses a message-signature pair.
Signature with Efficient Protocols

Signature Scheme with Efficient Protocols (Camenisch-Lysyanskya, SCN’02)

Signer

Verifier

Sign

Verify

Message

Signature

Message
Signature with Efficient Protocols

Signature Scheme with Efficient Protocols (Camenisch-Lysyanskya, SCN’02)

- Sign committed values

Flowchart showing the interaction between Signer and Verifier:
- Signer
  - Sign
  - Message
- Verifier
  - Verify
  - Message
  - Open
  - Signature
Signature with Efficient Protocols

Signature Scheme with Efficient Protocols (Camenisch-Lysyanskya, SCN’02)

Signer

Message

Verifier

Verify

Message

Open

Signature

PoK

ZKPoK

- Sign committed values
- Proof of Knowledge (PoK) of (Message; Signature)
## Lattice

A lattice is a discrete subgroup of $\mathbb{R}^n$. Can be seen as integer linear combinations of a finite set of vectors.
Lattice-Based Cryptography

### Lattice

A lattice is a discrete subgroup of $\mathbb{R}^n$. Can be seen as integer linear combinations of a finite set of vectors.

### Why?

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

→ Finding a non-zero short vector in a lattice is hard.
Hardness Assumptions: SIS and LWE
Parameters: $n$ dimension, $m \geq n$, $q$ modulus.
For $A \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$:

**Small Integer Solution**

<table>
<thead>
<tr>
<th>$x$</th>
<th>$A$</th>
</tr>
</thead>
</table>

$A = 0 \mod q$

**Learning With Errors**

\[
\begin{pmatrix}
A \\
m \\
n
\end{pmatrix},
\begin{pmatrix}
s \\
e
\end{pmatrix}
\]

$s \leftarrow \mathbb{Z}_q^n$, 
e a small error.

Goal: Given $A \leftarrow \mathbb{Z}_q^{m \times n}$, find $x \in \mathbb{Z}^m \setminus \{0\}$ small.

Goal: Given $(A, A s + e)$, find $s \in \mathbb{Z}_q^n$. 
Provable Security

Lattice hard problems
find a short vector in a lattice.
Worst-case

Hardness assumptions
LWE, SIS.
Average-case

Security properties
anonymity, traceability, non-frameability.
Group Signatures: History

1991 Chaum and Van Heyst: introduction

2000 Ateniese, Camenisch, Joye and Tsudik: first scalable solution

2003 Bellare, Micciancio and Warinschi: model for static groups
Group Signatures: History

1991  Chaum and Van Heyst: introduction

2000  Ateniese, Camenisch, Joye and Tsudik: first scalable solution

2003  Bellare, Micciancio and Warinschi: model for static groups

2004  Kiayias and Yung: model for dynamic groups

2004  Bellare, Shi and Zhang: model for dynamic groups
1991  Chaum and Van Heyst: introduction
2000  Ateniese, Camenisch, Joye and Tsudik: first scalable solution
2003  Bellare, Micciancio and Warinschi: model for static groups
2004  Kiayias and Yung: model for dynamic groups
2004  Bellare, Shi and Zhang: model for dynamic groups
2010  Gordon, Katz and Vaikuntanathan: first lattice-based scheme
2013  Laguillaumie, Langlois, Libert and Stehlé: sub-linear signatures
Group Signatures: History

1991  Chaum and Van Heyst: introduction

2000  Ateniese, Camenisch, Joye and Tsudik: first scalable solution

2003  Bellare, Micciancio and Warinschi: model for static groups

2004  Kiayias and Yung: model for dynamic groups

2004  Bellare, Shi and Zhang: model for dynamic groups

2010  Gordon, Katz and Vaikuntanathan: first lattice-based scheme

2013  Laguillaumie, Langlois, Libert and Stehlé: sub-linear signatures

No dynamic group signature scheme based on lattices
Outline

Introduction

Definition

Presentation of the Scheme

Conclusion
Signature with Efficient Protocols (CL'02)

A signature scheme \((\text{Keygen}, \text{Sign}_{sk}, \text{Verif}_{vk})\) with companion protocols:

- Sign a committed value;
- Prove possession of a signature.
Signature with Efficient Protocols (CL’02)

A signature scheme \((\text{Keygen}, \text{Sign}_{sk}, \text{Verif}_{vk})\) with companion protocols:

- Sign a committed value;
- Prove possession of a signature.

Security

- Unforgeability;
Signature with Efficient Protocols (CL'02)

A signature scheme \((\text{Keygen}, \text{Sign}_{sk}, \text{Verif}_{vk})\) with companion protocols:

- Sign a committed value;
- Prove possession of a signature.

Security

- Unforgeability;
- Security of the two protocols;
- Anonymity.
Signature with Efficient Protocols (CL'02)

A signature scheme \((\text{Keygen}, \text{Sign}_{sk}, \text{Verif}_{vk})\) with companion protocols:

- Sign a committed value;
- Prove possession of a signature.

### Security

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

→ many applications for privacy-based protocols
Signature with Efficient Protocols (CL'02)

A signature scheme \((\text{Keygen}, \text{Sign}_{sk}, \text{Verif}_{vk})\) with companion protocols:

- Sign a committed value;
- 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.*
Dynamic Group Signature

Keygen

Join \((\text{sec}_i, \text{cert}_i)\)

ok

\(Y\)
Dynamic Group Signature

Sign

\text{gsk}[d]

\text{Verify}

M, \Sigma

\gamma
Dynamic Group Signature

$M, \Sigma$
Dynamic Group Signature

\[ M, \Sigma \]

Open

\[ \text{ok} \]

\[ i \]
Dynamic Group Signature

**Dynamic Group Signature**

It is a tuple of algorithms (Setup, Join, Sign, Verify, Open) acting according to their names.
Dynamic Group Signature

It is a tuple of algorithms \((\text{Setup, Join, Sign, Verify, Open})\) acting according to their names.

- **Setup:**
  - Input: security parameter \(\lambda\), bound on group size \(N\)
  - Output: public parameters \(\mathcal{Y}\), group manager’s secret key \(S_{GM}\), the opening authority’s secret key \(S_{OA}\);
Dynamic Group Signature

It is a tuple of algorithms \((\text{Setup}, \text{Join}, \text{Sign}, \text{Verify}, \text{Open})\) acting according to their names.

- **Join**: interactive protocols between \(U_i \leftrightarrow GM\). Provide \((\text{cert}_i, \text{sec}_i)\) to \(U_i\). Where \(\text{cert}_i\) attests the secret \(\text{sec}_i\). Update the user list along with the certificates;
**Dynamic Group Signature**

It is a tuple of algorithms (**Setup**, **Join**, **Sign**, **Verify**, **Open**) acting according to their names.

- **Sign** and **Verify** proceed in the obvious way;
- **Open**:
  - Input: OA’s secret $S_{OA}$, $M$ and $\Sigma$
  - Output: $i$. 

<table>
<thead>
<tr>
<th>Dynamic Group Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>It is a tuple of algorithms (<strong>Setup</strong>, <strong>Join</strong>, <strong>Sign</strong>, <strong>Verify</strong>, <strong>Open</strong>) acting according to their names.</td>
</tr>
<tr>
<td>▶ <strong>Sign</strong> and <strong>Verify</strong> proceed in the obvious way;</td>
</tr>
<tr>
<td>▶ <strong>Open</strong>:</td>
</tr>
<tr>
<td>- Input: OA’s secret $S_{OA}$, $M$ and $\Sigma$</td>
</tr>
<tr>
<td>- Output: $i$.</td>
</tr>
</tbody>
</table>
Security

Three security notions

- **Anonymity**: only OA can open a signature;
Three security notions

- **Anonymity**: only OA can open a signature;

- **Traceability** (= security of honest GM against users):
  no coalition of malicious users can create a signature that cannot be traced to one of them;
Three security notions

- **Anonymity**: only OA can open a signature;

- **Traceability** (= security of honest GM against users): no coalition of malicious users can create a signature that cannot be traced to one of them;

- **Non-frameability** (= security of honest members): colluding GM and OA cannot frame honest users.
Outline

Introduction

Definition

Presentation of the Scheme

Conclusion
Based on a variant of Boyen’s signature (PKC'10)

Given \( A \in \mathbb{Z}_q^{n \times m} \) and \( \{ A_i \}_{i=0}^\ell \in \mathbb{Z}_q^{n \times m} \), the signature is a small \( d \in \mathbb{Z}^{2m} \) s.t.

\[
A_0 + \sum_{j=1}^{\ell} m_j A_j \cdot d = 0 \ [q].
\]

The private key is a short \( T_A \in \mathbb{Z}_q^{m \times m} \) s.t.

\[
A \cdot T_A = 0 \ [q].
\]
Signature with Efficient Protocols

Based on a variant of Boyen’s signature (PKC’10)

Given \( A \in \mathbb{Z}_q^{n \times m} \) and \( \{ A_i \}_{i=0}^\ell \in \mathbb{Z}_q^{n \times m} \), the signature is a small

\[
d \in \mathbb{Z}^{2m} \quad \text{s.t.} \quad A A_0 + \sum_{j=1}^\ell m_j A_j \cdot d = 0 \ [q].
\]

The private key is a short \( T_A \in \mathbb{Z}_q^{m \times m} \) such that

\[
A \cdot T_A = 0 \ [q].
\]

(A modification of) Böhl et al.’s variant (Eurocrypt’13)

\( \tau \leftarrow \mathcal{U}(\{0, 1\}^\ell) \), \( D \) and \( u \) are public, \( m \in \{0, 1\}^{2m} \) encodes \( \text{Msg} \).

\[
A A_0 + \sum_{j=1}^\ell \tau_j A_j \cdot d = u + D \cdot m \ [q].
\]

\( \rightarrow \quad \sigma = (\tau, d) \)
Our Signature with Efficient Protocols

To sign $M \in \{0, 1\}^{2m}$

- Sample random $\tau \in \{0, 1\}^\ell$, random $s \in D_{\mathbb{Z}^{2m}}$
- Compute $C_M = D_0 \cdot s + D_1 \cdot M \in \mathbb{Z}_q^{2n}$
Our Signature with Efficient Protocols

To sign $M \in \{0, 1\}^{2m}$

- Sample random $\tau \in \{0, 1\}^\ell$, random $s \in D_{\mathbb{Z}^{2m}, \bar{\sigma}}$
- Compute $C_M = D_0 \cdot s + D_1 \cdot M \in \mathbb{Z}^{2n}$
- Using $T_A$, sample a short $d$ s.t.

$$A = A_0 + \sum_{j=1}^\ell \tau_j \cdot A_j$$

$$\Sigma = (\tau, d, s) \in \{0, 1\}^\ell \times \mathbb{Z}^{2m} \times \mathbb{Z}^{2m}$$

$$\text{bin}(C_M) \quad (*)$$
Our Signature with Efficient Protocols

To sign $M \in \{0, 1\}^{2m}$

- Sample random $\tau \in \{0, 1\}^\ell$, random $s \in D_{\mathbb{Z}^{2m}, \bar{\sigma}}$
- Compute $C_M = D_0 \cdot s + D_1 \cdot M \in \mathbb{Z}^{2n}_q$
- Using $T_A$, sample a short $d$ s.t.

$$n \quad A_0 + \sum_{j=1}^{\ell} \tau_j \cdot A_j$$

$$2m \quad u + D$$

$$\bin(C_M) \quad (\ast)$$

$$\Sigma = (\tau, d, s) \in \{0, 1\}^\ell \times \mathbb{Z}^{2m} \times \mathbb{Z}^{2m}$$

To verify: check that $d$ is short and that $\Sigma$ satisfies $(\ast)$. 
Our Signature **with Efficient Protocols**

Kawachi *et al.* (Asiacrypt’08) commitment:

\[ C_M = D_0 \cdot s + D_1 \cdot M \]

Is already embedded in Böhl *et al.* signature.
Our Signature with Efficient Protocols

Kawachi et al. (Asiacrypt’08) commitment:

\[ C_M = D_0 \cdot s + D_1 \cdot M \]

Is already embedded in Böhl et al. signature.

**Difficulty**: In the proof, for one of the message, the signature has a different distribution.
Our Signature **with Efficient Protocols**

Kawachi *et al.* (Asiacrypt’08) commitment:

\[ C_M = D_0 \cdot s + D_1 \cdot M \]

Is already embedded in Böhl *et al.* signature.

**Difficulty**: In the proof, for one of the message, the signature has a different distribution.

**Solution**: Use Rényi divergence instead of statistical distance to bound adversary’s advantage [BLLSS15].
Rényi Divergence

Presentation

\[ R_a(P\|Q) = \left( \sum_{x \in \text{Supp}(P)} \frac{P(x)^a}{Q(x)^{a-1}} \right)^{1/(a-1)} \]
Rényi Divergence

Measurement of the distance between two distributions

\[ R_a(P \parallel Q) = \left( \sum_{x \in \text{Supp}(P)} \frac{P(x)^a}{Q(x)^{a-1}} \right)^{1/(a-1)} \]
Rényi Divergence

Presentation

\[ R_a(P \parallel Q) = \left( \sum_{x \in \text{Supp}(P)} \frac{P(x)^a}{Q(x)^{a-1}} \right)^{1/(a-1)} \]

- Measurement of the distance between two distributions
- Multiplicative instead of additive
  - Only use it once in the proof
Rényi Divergence

Presentation

\[ R_a(P\|Q) = \left( \sum_{x \in \text{Supp}(P)} \frac{P(x)^a}{Q(x)^{a-1}} \right)^{1/(a-1)} \]

- Measurement of the distance between two distributions
- Multiplicative instead of additive
  - Only use it once in the proof
- **Probability preservation:**

\[ Q(A) \geq P(A)^{\frac{a}{a-1}} / R_a(P\|Q) \]
Our Signature with efficient protocols

**Kawachi et al.** (Asiacrypt’08) commitment:

For \( D_0, D_1 \in \mathbb{Z}_q^{2n \times 2m} \), \( s \leftarrow D_{\mathbb{Z}_q^{2m}, \sigma}, M \in \{0, 1\}^{2m} \)

\[
C_M = D_0 \cdot s + D_1 \cdot M \ [q]
\]

Compatible with Stern’s protocol (Crypto’93, [LNSW; PKC’13])

\( \implies \) ZK proof compatible with the signature
Stern’s Protocol (Crypto’93)

**Stern’s protocol** is a ZK proof for Syndrome Decoding Problem.
Stern’s Protocol (Crypto’93)

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

**Syndrome Decoding Problem**

Given \( P \in \mathbb{Z}_2^{n \times m} \) and \( v \in \mathbb{Z}_2^n \), find \( x \) s.t. \( w(x) = w \) and \( Pm^n x = v \mod 2 \)
Stern’s Protocol (Crypto’93)

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

**Syndrome Decoding Problem**

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

\[
P \begin{bmatrix} n \\ m \end{bmatrix} \begin{bmatrix} P \\ x \end{bmatrix} = v \quad \text{mod} \; 2
\]

**[KTX08]**: mod 2 → mod q

**[LNSW13]**: Extend Stern’s protocol for SIS and LWE statements

Recent uses of Stern-like protocols in lattice-based crypto:

**[LNW15]**, **[LLNW16]**, **[LLN MW16]**
Unified Framework using Stern’s Protocol

**Problem**: protocols using Stern’s proofs build them “from scratch”.

[LNW15, LLNW16]
Unified Framework using Stern’s Protocol

**Problem**: protocols using Stern’s proofs build them “from scratch”.

[LNW15, LLNW16]

Provide a framework to construct ZKAoK:

- to prove knowledge of an \( \mathbf{x} \in \{ -1, 0, 1 \}^n \) of a special form verifying \( \mathbf{P} \cdot \mathbf{x} = \mathbf{v} \mod q \)
  - many lattice statements reduce to this
  - this captures various and complex statements
Unified Framework using Stern’s Protocol

**Problem:** protocols using Stern’s proofs build them “from scratch”. [LNW15, LLNW16]

Provide a framework to construct ZK AoK:

- to prove knowledge of an $x \in \{-1, 0, 1\}^n$ of a special form verifying $P \cdot x = v \mod q$
  - many lattice statements reduce to this
  - this captures various and complex statements

- that uses [LNSW13]’s decomposition-extension framework and is combinatoric in Stern’s protocol manner
From Static to Dynamic

- Designed from a recent static group signature proposed by Ling, Nguyen and Wang [LNW15];
From Static to Dynamic

- Designed from a recent static group signature proposed by Ling, Nguyen and Wang [LNW15];

- **Non-frameability** requires to introduce **non-homogeneous terms** in the SIS-based relations satisfied by membership certificates;
From Static to Dynamic

- Designed from a recent static group signature proposed by Ling, Nguyen and Wang [LNW15];

- Non-frameability requires to introduce non-homogeneous terms in the SIS-based relations satisfied by membership certificates;

- Other solutions [LLLS13, NZZ15] use membership certificates made of a complete basis. . .

  . . . which is problematic with non-homogeneous terms.
From Static to Dynamic

Difficulties

- Separate the secrets between OA and GM;
From Static to Dynamic

Difficulties

- Separate the secrets between OA and GM;

- Bind the user’s secret $z_i$ to a unique public syndrome $v_i = F \cdot z_i \in \mathbb{Z}_q^{4n}$ for some matrix $F \in \mathbb{Z}_q^{4n \times 4m}$;
From Static to Dynamic

Difficulties

- Separate the secrets between OA and GM;

- Bind the user’s secret $z_i$ to a unique public syndrome $v_i = F \cdot z_i \in \mathbb{Z}_q^{4n}$ for some matrix $F \in \mathbb{Z}_q^{4n \times 4m}$;

Use our signature scheme with efficient protocol:
From Static to Dynamic

Difficulties

- **Difficulty**: achieving security against *framing attacks*:
From Static to Dynamic

Difficulties

- **Difficulty**: achieving security against **framing attacks**:
  - i.e., even a dishonest **GM** cannot create signatures that open to honest users;
  - Users need a membership secret with a corresponding secret key;
  - GM must certify that public key.
From Static to Dynamic

Difficulties

- **Difficulty**: achieving security against framing attacks:
  - i.e., even a dishonest GM cannot create signatures that open to honest users;
  - Users need a membership secret with a corresponding secret key;
  - GM must certify that public key.

- Be secure against framing attacks without compromising previous security properties;
From Static to Dynamic Our solution

Setup:

\[ \mathcal{Y} = (A, \{A_i\}_{i=0}^\ell, B, D, D_0, D_1, F, u) \]

\[ \ell = \log(N) \text{ (e.g. } \ell = 30) \]
From Static to Dynamic Our solution

**Setup:**

group public key \( \mathcal{V} = (A, \{A_i\}_{i=0}^\ell, B, D, D_0, D_1, F, u) \)

\( \ell = \log(N) \) (e.g. \( \ell = 30 \))

**Join algorithm:**

\( \mathcal{U}_i \)  \hspace{2cm} \text{GM}
From Static to Dynamic Our solution

Setup:

group public key \( \mathcal{Y} = (A, \{A_i\}_{i=0}^{\ell}, B, D, D_0, D_1, F, u) \)

\( \ell = \log(N) \) (e.g. \( \ell = 30 \))

Join algorithm:

\[ \mathcal{U}_i \quad \text{GM} \]

\[ z_i \leftarrow \text{short vector in } \mathbb{Z}^{4m} \]

\[ v_i = F \cdot z_i \]
From Static to Dynamic  Our solution

Setup:

\[ Y = (A, \{A_i\}_{i=0}^\ell, B, D, D_0, D_1, F, u) \]

\[ \ell = \log(N) \text{ (e.g. } \ell = 30) \]

Join algorithm:

\[ \mathcal{U}_i \]

\[ z_i \leftarrow \text{short vector in } \mathbb{Z}^{4m} \]

\[ v_i = F \cdot z_i \]

\[ v_i \leftarrow \text{GM} \]

\[ \text{id}_i \leftarrow \text{identity } \in \{0, 1\}^\ell \]

\[ (d_i, s_i) \text{ is a signature under tag } \text{id}_i \]
From Static to Dynamic  Our solution

Setup:

\[ \mathcal{Y} = (A, \{A_i\}_{i=0}^\ell, B, D, D_0, D_1, F, u) \]

\[ \ell = \log(N) \text{ (e.g. } \ell = 30) \]

Join algorithm:

\[ U_i \]

\[ z_i \leftarrow \text{short vector in } \mathbb{Z}^{4m} \]

\[ v_i = F \cdot z_i \]

\[ \text{If } (id_i, d_i, s_i) \text{ does not verify, abort} \]

\[ (sec_i; cert_i) = (z_i; (id_i, d_i, s_i)) \]
From Static to Dynamic Our solution — further steps

<table>
<thead>
<tr>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCA-Anonymity: anonymity under opening oracle.</td>
</tr>
</tbody>
</table>
### Goal

<table>
<thead>
<tr>
<th>CCA-Anonymity: anonymity under opening oracle.</th>
</tr>
</thead>
</table>

↑

<table>
<thead>
<tr>
<th>Canetti-Halevi-Katz transformation (Eurocrypt’04)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Any IBE implies \textit{IND-CCA}–secure public key encryption.</th>
</tr>
</thead>
</table>
From Static to Dynamic  Our solution — further steps

<table>
<thead>
<tr>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCA-Anonymity: anonymity under opening oracle.</td>
</tr>
</tbody>
</table>

Canetti-Halevi-Katz transformation  (Eurocrypt’04)

Any IBE implies *IND-CCA*-secure public key encryption.

<table>
<thead>
<tr>
<th>Identity Based Encryption  (Shamir’84, Boneh-Franklin’01)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encryption computes  ( C \leftarrow \text{Enc}(MPK, ID, M) )</td>
</tr>
<tr>
<td>Decryption computes  ( M \leftarrow \text{Dec}(MPK, C, d_{ID}) ) where  ( d_{ID} \leftarrow \text{Keygen}(MSK, ID) )</td>
</tr>
</tbody>
</table>
From Static to Dynamic Our solution

Sign algorithm:
$c := \text{Enc}(v_i)$
From Static to Dynamic Our solution

**Sign** algorithm:

\[ c := \text{Enc}(v_i) \quad \pi_K := \text{proof that } c \text{ is correct and that} \]

\[ A_0 + \sum_{j=1}^{\ell} \text{id}_j \cdot A_j = u + D \]

\[ \text{bin}(C_{v_i}) \]
From Static to Dynamic

Our solution

**Sign algorithm:**

\[ c := \text{Enc}(v_i) \quad \pi_K := \text{proof that } c \text{ is correct and that} \]

\[
\begin{align*}
A & \quad A_0 + \sum_{j=1}^{\ell} \text{id}_j \cdot A_j \\
\text{d} & \quad = \\
\text{u} & \quad + \\
\text{D} & \quad \text{bin}(C_{v_i})
\end{align*}
\]

Where is the message? [BSZ04]

Inside \( \pi_K \), encoded in the Fiat-Shamir transformation from ZK-proofs to NIZK-proofs.
From Static to Dynamic Our solution

Verify algorithm:

- A user verifies if $\pi_K$ is correct.
From Static to Dynamic: Our solution

**Verify algorithm:**

- A user verifies if $\pi_K$ is correct.

**Open algorithm:**

- \textbf{OA} decrypts $c$ to get $v_i$;
- \textbf{OA} searches for the associated $i$ in the Join transcripts, and if so, returns $i$, otherwise abort.
### Group Signatures: Comparative Table

<table>
<thead>
<tr>
<th>Scheme</th>
<th>LLLS</th>
<th>NZZ</th>
<th>LNW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group PK</td>
<td>$\tilde{O}(\lambda^2) \cdot \log N_{gs}$</td>
<td>$\tilde{O}(\lambda^2)$</td>
<td>$\tilde{O}(\lambda^2) \cdot \log N_{gs}$</td>
</tr>
<tr>
<td>User’s SK</td>
<td>$\tilde{O}(\lambda^2)$</td>
<td>$\tilde{O}(\lambda^2)$</td>
<td>$\tilde{O}(\lambda)$</td>
</tr>
<tr>
<td>Signature</td>
<td>$\tilde{O}(\lambda) \cdot \log N_{gs}$</td>
<td>$\tilde{O}(\lambda + \log^2 N_{gs})$</td>
<td>$\tilde{O}(\lambda) \cdot \log N_{gs}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scheme</th>
<th>LLNW</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group PK</td>
<td>$\tilde{O}(\lambda^2)$</td>
<td>$\tilde{O}(\lambda^2) \cdot \log N_{gs}$</td>
</tr>
<tr>
<td>User’s SK</td>
<td>$\tilde{O}(\lambda) \cdot \log N_{gs}$</td>
<td>$\tilde{O}(\lambda)$</td>
</tr>
<tr>
<td>Signature</td>
<td>$\tilde{O}(\lambda) \cdot \log N_{gs}$</td>
<td>$\tilde{O}(\lambda) \cdot \log N_{gs}$</td>
</tr>
</tbody>
</table>
Outline

Introduction

Definition

Presentation of the Scheme

Conclusion
Conclusion

Main Contributions:

- Lattice-based signature with efficient protocols;
  - for obtaining signatures on committed message
  - for proving possession of a message-signature pair
- First dynamic group signature based on lattice assumptions;
- Unified framework for proving modular linear equations using Stern’s technique.

Technical contributions:

- Combine Böhl et al. signature + Ling et al. ZK proofs
  \(\Rightarrow\) signature with efficient protocols;
- A method of signing public keys so that knowledge of the secret key can be efficiently proved.
Thank you all for your attention!
One-Time Signature

Definition

A one-time signature scheme consists of a triple of algorithms $\Pi^{\text{ots}} = (G, S, V)$. Behaves like a digital signature scheme.

Strong unforgeability: impossible to forge a valid signature even for a previously signed message.

Usage

We use one-time signature to provide CCA anonymity using Canetti-Halevi-Katz methodology.
**CCA anonymity**

### Definition

No PPT adversary $\mathcal{A}$ can win the following game with non-negligible probability:

- $\mathcal{A}$ makes open queries.
- $\mathcal{A}$ chooses $M^*$ and two different $(\text{cert}_i^*, \text{sec}_i^*)_{i \in \{0,1\}}$
- $\mathcal{A}$ receives $\sigma^* = \text{Sign}_{\text{cert}_b^*, \text{sec}_b^*}(M^*)$ for some $b \in \{0,1\}$
- $\mathcal{A}$ makes other open queries
- $\mathcal{A}$ returns $b'$, and wins if $b = b'$
ZK Proofs

**Σ-protocol [Dam10]**

3-move scheme: \((\text{Commit}, \text{Challenge}, \text{Answer})\) between 2 users.

**Fiat-Shamir Heuristic**

Make the Σ-protocol **non-interactive** by setting the challenge to be \(H(\text{Commit}, \text{Public})\)
From Static to Dynamic Our solution – Ingredients

Security proof of the Boyen signature

Lattice algorithms use short basis as *trapdoor* information.

**SampleUp** \[ A' = \begin{bmatrix} A \\ B \cdot A + C \end{bmatrix} \in \mathbb{Z}_q^{2m \times n}, A \in \mathbb{Z}_q^{m \times n}, T_A \in \mathbb{Z}_q^{m \times m}, \sigma \mapsto \text{gaussian } v \in \mathbb{Z}_q^n, \text{ s.t. } v^T A' = 0[q] \]

**SampleDown** \[ A' = \begin{bmatrix} A \\ B \cdot A + C \end{bmatrix} \in \mathbb{Z}_q^{2m \times n}, C \in \mathbb{Z}_q^{m \times n}, T_C \in \mathbb{Z}_q^{m \times m}, \sigma \mapsto \text{gaussian } v \in \mathbb{Z}_q^n, \text{ s.t. } v^T A' = 0[q] \]
From Static to Dynamic Our solution – Ingredients

Security proof of the Boyen signature

Boyen’s signature

\[ d^T \begin{bmatrix} \frac{A}{A_0 + \sum_{i=1}^{\ell} m_i A_i} \end{bmatrix} = 0[q] \]

Idea. Set \( A_i = Q_i A + h_i C \)

\[ \rightarrow \begin{bmatrix} \frac{A}{A_0 + \sum_{i=1}^{\ell} m_i A_i} \end{bmatrix} = \begin{bmatrix} \frac{A}{(Q_0 + \sum_{i=1}^{\ell} m_i Q_i) A + h_M C} \end{bmatrix} \]

⇒ We can use SampleUp in the real setup and SampleDown in the reduction whenever \( h_M \neq 0 \).
From Static to Dynamic Our solution – Ingredients

Security proof of the Boyen signature

Recall

$$A' := \left[ \begin{array}{c} A \\ A_0 + \sum_{i=1}^{\ell} m_i A_i \end{array} \right] = \left[ \begin{array}{c} A \\ (Q_0 + \sum_{i=1}^{\ell} m_i Q_i)A + h_M C \end{array} \right]$$

**Forgery.** $A$ outputs $d^* = [d_1^* T | d_2^* T]^T$ and $M^* = m_1^* \ldots m_\ell^*$ such that $d^*^T A' = 0$.

If $h_{M^*} = 0$, then

$$\left( d_1^* T + d_2^* T \left( Q_0 + \sum_{i=1}^{\ell} m_i^* Q_i \right) \right) A = 0[q]$$

valid SIS solution
Remark

Boyen’s signature: the reduction aborts if $C$ vanishes.
Böhl et al.: answer the request by “programming” the vector

$$u^T = d^T \left[ \frac{A}{(Q_0 + \sum_{i=1}^{\ell} m_i^T Q_i)A} \right] - z_i^T D.$$ 

Problem

In this request, a sum of two discrete gaussian is generated differently from the real Join protocol.
$\Rightarrow$ Not the same standard deviation.
From Static to Dynamic  

Our solution

Problem

\[ z_{i,0}, z_{i,1}, z_i \in \mathbb{Z}^m \]

Consequence.

\[
\{(z_i, z_{i,0}, z_{i,1}) | z_{i,0} \leftarrow D_{\sigma_0}, z_{i,1} \leftarrow D_{\sigma_1}, z_i = z_{i,0} + z_{i,1}\} \\
\sim \Delta \\
\{(z_i, z_{i,0}, z_{i,1}) | z_i \leftarrow D_{\sigma}, z_{i,0} \leftarrow D_{\sigma_0}, z_{i,1} = z_i - z_{i,0}\} 
\]
Rényi Divergence

Presentation

\[ R_a(P \Vert Q) = \left( \sum_{x \in \text{Supp}(P)} \frac{P(x)^a}{Q(x)^{a-1}} \right)^{1/(a-1)} \]
Rényi Divergence

Presentation

\[ R_a(P \parallel Q) = \left( \sum_{x \in \text{Supp}(P)} \frac{P(x)^a}{Q(x)^{a-1}} \right)^{1/(a-1)} \]

- Measurement of the distance between two distributions
Rényi Divergence

Measurement of the distance between two distributions

- Multiplicative instead of additive
- Probability preservation:

\[ Q(A) \geq P(A)^{\frac{a}{a-1}} / R_a(P \parallel Q) \]
Rényi Divergence

Hybrid argument:

Real game $\rightarrow$ Game 1 $\rightarrow$ Game 2 $\rightarrow$ Hard Game

- Hardness assumptions -

Bound winning probability.
Can be done through probability preservation!

Recall

$$Q(A) \geq P(A) \frac{a}{a-1} / R_a(P \parallel Q)$$

$$\Pr[W_2] \geq \Pr[W_1] \frac{a}{a-1} / R_a(Game_1 \parallel Game_2)$$

For instance: $$\Pr[W_2] \geq \Pr[W_1]^2 / R_2(Game_1 \parallel Game_2)$$
Rényi Divergence
In Crypto

**Consequence**

Usually use *statistical distance* to measure distance between probabilities.

- In our setting, implies $q \sim \exp(\lambda)$ (**smudging**)

- Higher cost compared to usual lattice-based crypto parameters