Some Recent Uses of Stern-like Protocols in Lattice-Based Cryptography

Khoa Nguyen, Nanyang Technological University

Quantum-Safe Crypto Workshop, CQT, NUS

October, 3rd, 2016

Outline of the Talk

1. Introduction
   - Zero-Knowledge Protocols
   - Zero-Knowledge Protocols in Lattice-Based Cryptography
2. Stern-like Protocols
   - Stern-KTX Protocol
   - Abstracting Stern's Protocol
   - Techniques and Applications
3. Conclusion

Zero-Knowledge Protocols

[Goldwasser-Micali-Rackoff 1985]

Prover		**Verifier**
"The problem has been solved!" ... but I won't reveal my solution 🙂		"Really? Convince me!" 🤔
	⟶	
	⟵	
	......	
	⟶	✓ OK

- *Zero-knowledge:* $\mathcal{V}$ learns nothing except the validity of the statement.
- *Soundness:* Dishonest $\mathcal{P}$ should *not* be able to cheat.
- *Completeness:* Honest $\mathcal{P}$ should be able to convince $\mathcal{V}$.

Numerous applications: identification, signatures, anonymity schemes, MPC, ...

Existing ZK Protocols in Lattice-Based Crypto

- Lattice-based crypto [Ajtai'96, GPV'08]: An exciting area.
- Existing zero-knowledge protocols in LBC belong to 2 main classes.

Existing ZK Protocols in Lattice-Based Crypto

- Lattice-based crypto [Ajtai'96, GPV'08]: An exciting area.
- Existing zero-knowledge protocols in LBC belong to 2 main classes.

1. Schnorr-like [Schnorr'89] approach. Technique: **masking**.
   - Pioneered by Lyubashevsky [L'08,12].
   - Additional technique: **rejection sampling**.
   - Relatively efficient, imperfect completeness, extraction gap.

Existing ZK Protocols in Lattice-Based Crypto

- Lattice-based crypto [Ajtai'96, GPV'08]: An exciting area.
- Existing zero-knowledge protocols in LBC belong to 2 main classes.

1. Schnorr-like [Schnorr'89] approach. Technique: **masking**.
   - Pioneered by Lyubashevsky [L'08,12].
   - Additional technique: **rejection sampling**.
   - Relatively efficient, imperfect completeness, extraction gap.
2. Stern-like [Stern'93,96] approach. Technique: **permuting**, **masking**.
   - First used by Kawachi et al. [KTX'08]: restricted relation.
   - Additional techniques [LNSW'13]: **decomposition** and **extension**.
   - Recently developed into a strong tool for privacy-preserving LBC.
   - Less efficient, perfect completeness, no extraction gap (i.e., the exact constraints of prover's secret are "captured").

A Brief Story of Stern-like Protocols for LBC

Stern's original protocol: SD relation (naturally appearing in code-based crypto):

(public matrix) $\cdot$ (binary secret vector with fixed weight) $=$ (public vector) mod 2.

A Brief Story of Stern-like Protocols for LBC

Stern's original protocol: SD relation (naturally appearing in code-based crypto):

(public matrix) $\cdot$ (binary secret vector with fixed weight) $=$ (public vector) mod 2.

- [KTX'08] shifted Stern's to mod q $\Rightarrow$ a restricted, unnatural version of ISIS.

A Brief Story of Stern-like Protocols for LBC

Stern's original protocol: SD relation (naturally appearing in code-based crypto):

(public matrix) · (binary secret vector with fixed weight) = (public vector) mod 2.

- [KTX'08] shifted Stern's to mod q $\Rightarrow$ a restricted, unnatural version of ISIS.
- [LNSW'13]'s **decomposition-extension** techniques $\Rightarrow$ (I)SIS and LWE.
  $\Rightarrow$ immediate applications: (ID-based) identifications, plaintext knowledge.

A Brief Story of Stern-like Protocols for LBC

Stern's original protocol: SD relation (naturally appearing in code-based crypto):

(public matrix) $\cdot$ (binary secret vector with fixed weight) $=$ (public vector) mod 2.

- [KTX'08] shifted Stern's to mod q $\Rightarrow$ a restricted, unnatural version of ISIS.
- [LNSW'13]'s **decomposition-extension** techniques $\Rightarrow$ (I)SIS and LWE.
  $\Rightarrow$ immediate applications: (ID-based) identifications, plaintext knowledge.
- [LLNW'14]: a hidden (μ, σ) pair for the Bonsai signature [CHKP'10].
  $\Rightarrow$ group signatures, policy-based signatures [CNW'15].

A Brief Story of Stern-like Protocols for LBC

Stern's original protocol: SD relation (naturally appearing in code-based crypto):

(public matrix) · (binary secret vector with fixed weight) = (public vector) mod 2.

- [KTX'08] shifted Stern's to mod q ⇒ a restricted, unnatural version of ISIS.
- [LNSW'13]'s **decomposition-extension** techniques ⇒ (I)SIS and LWE.
  ⇒ immediate applications: (ID-based) identifications, plaintext knowledge.
- [LLNW'14]: a hidden (μ, σ) pair for the Bonsai signature [CHKP'10].
  ⇒ group signatures, policy-based signatures [CNW'15].
- [LNW'15]: a hidden (μ, σ) pair for the Boyen signature [Boy'10].

A Brief Story of Stern-like Protocols for LBC

Stern's original protocol: SD relation (naturally appearing in code-based crypto):

(public matrix) · (binary secret vector with fixed weight) = (public vector) mod 2.

- [KTX'08] shifted Stern's to mod q ⇒ a restricted, unnatural version of ISIS.
- [LNSW'13]'s **decomposition-extension** techniques ⇒ (I)SIS and LWE.
  ⇒ immediate applications: (ID-based) identifications, plaintext knowledge.
- [LLNW'14]: a hidden (μ, σ) pair for the Bonsai signature [CHKP'10].
  ⇒ group signatures, policy-based signatures [CNW'15].
- [LNW'15]: a hidden (μ, σ) pair for the Boyen signature [Boy'10].
- [LLNW'16]: a hidden path in a Merkle hash tree ⇒ logarithmic-size set membership ⇒ logarithmic-size ring signatures, trapdoor-free GS.

A Brief Story of Stern-like Protocols for LBC

Stern's original protocol: SD relation (naturally appearing in code-based crypto):

(public matrix) · (binary secret vector with fixed weight) = (public vector) mod 2.

- [KTX'08] shifted Stern's to mod q ⇒ a restricted, unnatural version of ISIS.
- [LNSW'13]'s **decomposition-extension** techniques ⇒ (I)SIS and LWE.
  ⇒ immediate applications: (ID-based) identifications, plaintext knowledge.
- [LLNW'14]: a hidden (μ, σ) pair for the Bonsai signature [CHKP'10].
  ⇒ group signatures, policy-based signatures [CNW'15].
- [LNW'15]: a hidden (μ, σ) pair for the Boyen signature [Boy'10].
- [LLNW'16]: a hidden path in a Merkle hash tree ⇒ logarithmic-size set membership ⇒ logarithmic-size ring signatures, trapdoor-free GS.
- [LLMNW'16-(1)]: **abstracting Stern**; the μ being signed is U's hidden pk.
  ⇒ "signatures with protocols", dynamic GS, anonymous credentials.

A Brief Story of Stern-like Protocols for LBC

Stern's original protocol: SD relation (naturally appearing in code-based crypto):

(public matrix) · (binary secret vector with fixed weight) = (public vector) mod 2.

- [KTX'08] shifted Stern's to mod q ⇒ a restricted, unnatural version of ISIS.
- [LNSW'13]'s **decomposition-extension** techniques ⇒ (I)SIS and LWE.
  ⇒ immediate applications: (ID-based) identifications, plaintext knowledge.
- [LLNW'14]: a hidden (μ, σ) pair for the Bonsai signature [CHKP'10].
  ⇒ group signatures, policy-based signatures [CNW'15].
- [LNW'15]: a hidden (μ, σ) pair for the Boyen signature [Boy'10].
- [LLNW'16]: a hidden path in a Merkle hash tree ⇒ logarithmic-size set membership ⇒ logarithmic-size ring signatures, trapdoor-free GS.
- [LLMNW'16-(1)]: **abstracting Stern**; the μ being signed is U's hidden pk.
  ⇒ "signatures with protocols", dynamic GS, anonymous credentials.
- [LLMNW'16-(2)]: "quadratic relations", i.e., (secret matrix) · (secret vector)
  ⇒ group encryption.

Outline

Stern's Original Protocol

Parameters: $k, d, t \in \mathbb{Z}$; $\mathcal{B} = \{\mathbf{w} \in \{0,1\}^d : wt(\mathbf{w}) = t\}$; COM.

Common input: Matrix $\mathbf{M} \in \mathbb{Z}_2^{k \times d}$, vector $\mathbf{v} \in \mathbb{Z}_2^k$.

$\mathcal{P}$'s goal: Proving knowledge of $\mathbf{w} \in \mathcal{B}$ s.t. $\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod 2$.

Prover **Verifier**

Stern's Original Protocol

Parameters: $k, d, t \in \mathbb{Z}$; $\mathcal{B} = \{\mathbf{w} \in \{0,1\}^d : wt(\mathbf{w}) = t\}$; COM.

Common input: Matrix $\mathbf{M} \in \mathbb{Z}_2^{k \times d}$, vector $\mathbf{v} \in \mathbb{Z}_2^k$.

$\mathcal{P}$'s goal: Proving knowledge of $\mathbf{w} \in \mathcal{B}$ s.t. $\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod 2$.

Prover | **Verifier**

1. Pick $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_2^d$, $\pi \xleftarrow{\$} \mathcal{S}_d$. Send $(\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$, where

$$\begin{cases} \mathbf{c}_1 = \mathrm{COM}\big(\pi, \mathbf{M} \cdot \mathbf{r}\big) \\ \mathbf{c}_2 = \mathrm{COM}\big(\pi(\mathbf{r})\big) \\ \mathbf{c}_3 = \mathrm{COM}\big(\pi(\mathbf{w} + \mathbf{r})\big) \end{cases}$$

Stern's Original Protocol

Parameters: $k, d, t \in \mathbb{Z}$; $\mathcal{B} = \{\mathbf{w} \in \{0,1\}^d : wt(\mathbf{w}) = t\}$; COM.

Common input: Matrix $\mathbf{M} \in \mathbb{Z}_2^{k \times d}$, vector $\mathbf{v} \in \mathbb{Z}_2^k$.

$\mathcal{P}$'s goal: Proving knowledge of $\mathbf{w} \in \mathcal{B}$ s.t. $\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod 2$.

Prover

1. Pick $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_2^d$, $\pi \xleftarrow{\$} \mathcal{S}_d$. Send $(\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$, where

$$\begin{cases} \mathbf{c}_1 = \mathrm{COM}(\pi, \mathbf{M} \cdot \mathbf{r}) \\ \mathbf{c}_2 = \mathrm{COM}(\pi(\mathbf{r})) \\ \mathbf{c}_3 = \mathrm{COM}(\pi(\mathbf{w} + \mathbf{r})) \end{cases}$$

Verifier

2. Send a challenge

$$Ch \xleftarrow{\$} \{1, 2, 3\}$$

Stern's Original Protocol

Parameters: $k, d, t \in \mathbb{Z}$; $\mathcal{B} = \{\mathbf{w} \in \{0,1\}^d : wt(\mathbf{w}) = t\}$; COM.

Common input: Matrix $\mathbf{M} \in \mathbb{Z}_2^{k \times d}$, vector $\mathbf{v} \in \mathbb{Z}_2^k$.

$\mathcal{P}$'s goal: Proving knowledge of $\mathbf{w} \in \mathcal{B}$ s.t. $\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod 2$.

Prover

Verifier

1. Pick $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_2^d$, $\pi \xleftarrow{\$} \mathcal{S}_d$. Send $(\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$, where

$$\begin{cases} \mathbf{c}_1 = \mathrm{COM}(\pi, \mathbf{M} \cdot \mathbf{r}) \\ \mathbf{c}_2 = \mathrm{COM}(\pi(\mathbf{r})) \\ \mathbf{c}_3 = \mathrm{COM}(\pi(\mathbf{w} + \mathbf{r})) \end{cases}$$

2. Send a challenge

$$Ch \xleftarrow{\$} \{1, 2, 3\}$$

3. If $Ch = 1$, reveal $\mathbf{c}_2$ and $\mathbf{c}_3$. Send $\pi(\mathbf{w})$ and $\pi(\mathbf{r})$

Stern's Original Protocol

Parameters: $k, d, t \in \mathbb{Z}$; $\mathcal{B} = \{\mathbf{w} \in \{0,1\}^d : wt(\mathbf{w}) = t\}$; COM.

Common input: Matrix $\mathbf{M} \in \mathbb{Z}_2^{k \times d}$, vector $\mathbf{v} \in \mathbb{Z}_2^k$.

$\mathcal{P}$'s goal: Proving knowledge of $\mathbf{w} \in \mathcal{B}$ s.t. $\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod 2$.

Prover

1. Pick $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_2^d$, $\pi \xleftarrow{\$} \mathcal{S}_d$. Send $(\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$, where

$$\begin{cases} \mathbf{c}_1 = \mathrm{COM}(\pi, \mathbf{M} \cdot \mathbf{r}) \\ \mathbf{c}_2 = \mathrm{COM}(\pi(\mathbf{r})) \\ \mathbf{c}_3 = \mathrm{COM}(\pi(\mathbf{w} + \mathbf{r})) \end{cases}$$

Verifier

2. Send a challenge

$$Ch \xleftarrow{\$} \{1, 2, 3\}$$

3. If $Ch = 1$, reveal $\mathbf{c}_2$ and $\mathbf{c}_3$. Send $\pi(\mathbf{w})$ and $\pi(\mathbf{r})$

Check that $\pi(\mathbf{w}) \in \mathcal{B}$ and

$$\begin{cases} \mathbf{c}_2 = \mathrm{COM}(\pi(\mathbf{r})) \\ \mathbf{c}_3 = \mathrm{COM}(\pi(\mathbf{w}) + \pi(\mathbf{r})) \end{cases}$$

Stern's Original Protocol

Parameters: $k, d, t \in \mathbb{Z}$; $\mathcal{B} = \{\mathbf{w} \in \{0,1\}^d : wt(\mathbf{w}) = t\}$; COM.

Common input: Matrix $\mathbf{M} \in \mathbb{Z}_2^{k \times d}$, vector $\mathbf{v} \in \mathbb{Z}_2^k$.

$\mathcal{P}$'s goal: Proving knowledge of $\mathbf{w} \in \mathcal{B}$ s.t. $\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod 2$.

Prover

Verifier

1. Pick $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_2^d$, $\pi \xleftarrow{\$} \mathcal{S}_d$. Send $(\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$, where

$$\begin{cases} \mathbf{c}_1 = \mathrm{COM}(\pi, \mathbf{M} \cdot \mathbf{r}) \\ \mathbf{c}_2 = \mathrm{COM}(\pi(\mathbf{r})) \\ \mathbf{c}_3 = \mathrm{COM}(\pi(\mathbf{w} + \mathbf{r})) \end{cases}$$

2. Send a challenge

$$Ch \xleftarrow{\$} \{1, 2, 3\}$$

3. If $Ch = 2$, reveal $\mathbf{c}_1$ and $\mathbf{c}_3$. Send π and $\mathbf{w} + \mathbf{r}$.

Stern's Original Protocol

Parameters: $k, d, t \in \mathbb{Z}$; $\mathcal{B} = \{\mathbf{w} \in \{0,1\}^d : wt(\mathbf{w}) = t\}$; COM.

Common input: Matrix $\mathbf{M} \in \mathbb{Z}_2^{k \times d}$, vector $\mathbf{v} \in \mathbb{Z}_2^k$.

$\mathcal{P}$'s goal: Proving knowledge of $\mathbf{w} \in \mathcal{B}$ s.t. $\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod 2$.

Prover

1. Pick $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_2^d$, $\pi \xleftarrow{\$} \mathcal{S}_d$. Send $(\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$, where

$$\begin{cases} \mathbf{c}_1 = \mathrm{COM}(\pi, \mathbf{M} \cdot \mathbf{r}) \\ \mathbf{c}_2 = \mathrm{COM}(\pi(\mathbf{r})) \\ \mathbf{c}_3 = \mathrm{COM}(\pi(\mathbf{w} + \mathbf{r})) \end{cases}$$

Verifier

2. Send a challenge

$$Ch \xleftarrow{\$} \{1, 2, 3\}$$

3. If $Ch = 2$, reveal $\mathbf{c}_1$ and $\mathbf{c}_3$. Send π and $\mathbf{w} + \mathbf{r}$.

Check that

$$\begin{cases} \mathbf{c}_1 = \mathrm{COM}(\pi, \mathbf{M} \cdot (\mathbf{w} + \mathbf{r}) - \mathbf{v}) \\ \mathbf{c}_3 = \mathrm{COM}(\pi(\mathbf{w} + \mathbf{r})) \end{cases}$$

Stern's Original Protocol

Parameters: $k, d, t \in \mathbb{Z}$; $\mathcal{B} = \{\mathbf{w} \in \{0,1\}^d : wt(\mathbf{w}) = t\}$; COM.

Common input: Matrix $\mathbf{M} \in \mathbb{Z}_2^{k \times d}$, vector $\mathbf{v} \in \mathbb{Z}_2^k$.

$\mathcal{P}$'s goal: Proving knowledge of $\mathbf{w} \in \mathcal{B}$ s.t. $\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod 2$.

Prover

1. Pick $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_2^d$, $\pi \xleftarrow{\$} \mathcal{S}_d$. Send $(\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$, where

$$\begin{cases} \mathbf{c}_1 = \mathrm{COM}(\pi, \mathbf{M} \cdot \mathbf{r}) \\ \mathbf{c}_2 = \mathrm{COM}(\pi(\mathbf{r})) \\ \mathbf{c}_3 = \mathrm{COM}(\pi(\mathbf{w} + \mathbf{r})) \end{cases}$$

Verifier

2. Send a challenge

$$Ch \xleftarrow{\$} \{1, 2, 3\}$$

3. If $Ch = 3$, reveal $\mathbf{c}_1$ and $\mathbf{c}_2$. Send π and $\mathbf{r}$.

Stern's Original Protocol

Parameters: $k, d, t \in \mathbb{Z}$; $\mathcal{B} = \{\mathbf{w} \in \{0,1\}^d : wt(\mathbf{w}) = t\}$; COM.

Common input: Matrix $\mathbf{M} \in \mathbb{Z}_2^{k \times d}$, vector $\mathbf{v} \in \mathbb{Z}_2^k$.

$\mathcal{P}$'s goal: Proving knowledge of $\mathbf{w} \in \mathcal{B}$ s.t. $\mathbf{M} \cdot \mathbf{w} = \mathbf{v}$ mod 2.

Prover

1. Pick $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_2^d$, $\pi \xleftarrow{\$} \mathcal{S}_d$. Send $(\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$, where

$$\begin{cases} \mathbf{c}_1 = \mathrm{COM}(\pi, \mathbf{M} \cdot \mathbf{r}) \\ \mathbf{c}_2 = \mathrm{COM}(\pi(\mathbf{r})) \\ \mathbf{c}_3 = \mathrm{COM}(\pi(\mathbf{w} + \mathbf{r})) \end{cases}$$

3. If $Ch = 3$, reveal $\mathbf{c}_1$ and $\mathbf{c}_2$. Send π and $\mathbf{r}$.

Verifier

2. Send a challenge

$$Ch \xleftarrow{\$} \{1, 2, 3\}$$

Check that

$$\begin{cases} \mathbf{c}_1 = \mathrm{COM}(\pi, \mathbf{M} \cdot \mathbf{r}) \\ \mathbf{c}_2 = \mathrm{COM}(\pi(\mathbf{r})) \end{cases}$$

Stern's Ideas and Kawachi et al.'s Adaptation

Why Stern's ideas work?

1. **Permuting**
   - $\mathbf{w} \in \mathcal{B} \iff \pi(\mathbf{w}) \in \mathcal{B}$;
   - $\mathbf{w} \in \mathcal{B}$ and $\pi \xleftarrow{\$} \mathcal{S}_d \longrightarrow \pi(\mathbf{w})$ is uniform in $\mathcal{B}$.

Stern's Ideas and Kawachi et al.'s Adaptation

Why Stern's ideas work?

1. **Permuting**
   - $\mathbf{w} \in \mathcal{B} \iff \pi(\mathbf{w}) \in \mathcal{B}$;
   - $\mathbf{w} \in \mathcal{B}$ and $\pi \xleftarrow{\$} \mathcal{S}_d \longrightarrow \pi(\mathbf{w})$ is uniform in $\mathcal{B}$.
2. **Masking** $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_2^d \longrightarrow \mathbf{w} + \mathbf{r}$ is uniform in $\mathbb{Z}_2^d$.

Stern's Ideas and Kawachi et al.'s Adaptation

Why Stern's ideas work?

1. **Permuting**
   - $\mathbf{w} \in \mathcal{B} \iff \pi(\mathbf{w}) \in \mathcal{B}$;
   - $\mathbf{w} \in \mathcal{B}$ and $\pi \xleftarrow{\$} \mathcal{S}_d \longrightarrow \pi(\mathbf{w})$ is uniform in $\mathcal{B}$.

2. **Masking** $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_2^d \longrightarrow \mathbf{w} + \mathbf{r}$ is uniform in $\mathbb{Z}_2^d$.

Kawachi et al.'s adaptation [KTX'08] to lattice setting:

- $\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod q$ and $\mathbf{w} \in \mathcal{B}$.
- Limited applications: identification, signature (via [FS'86]).

Stern's Ideas and Kawachi et al.'s Adaptation

Why Stern's ideas work?

1. **Permuting**
   - $\mathbf{w} \in \mathcal{B} \iff \pi(\mathbf{w}) \in \mathcal{B}$;
   - $\mathbf{w} \in \mathcal{B}$ and $\pi \xleftarrow{\$} \mathcal{S}_d \longrightarrow \pi(\mathbf{w})$ is uniform in $\mathcal{B}$.
2. **Masking** $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_2^d \longrightarrow \mathbf{w} + \mathbf{r}$ is uniform in $\mathbb{Z}_2^d$.

Kawachi et al.'s adaptation [KTX'08] to lattice setting:

- $\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod q$ and $\mathbf{w} \in \mathcal{B}$.
- Limited applications: identification, signature (via [FS'86]).

In lattice-based crypto, we usually work with

- $\mathbf{w} \xleftarrow{\$} \{0,1\}^d$ (no restriction on Hamming weight).
- $\mathbf{w} \xleftarrow{\$} [0, \beta]^d$ for some $1 \ll \beta \ll q$.
- Gaussian $\mathbf{w} \in [-\beta, \beta]^d$.

Abstracting Stern's Protocol

Suppose we want to use Stern to prove $\mathbf{w} \in \mathsf{VALID} \subset \mathbb{Z}^d$ s.t. $\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod q$.

Question: Which properties of VALID do we need?

Abstracting Stern's Protocol

Suppose we want to use Stern to prove $\mathbf{w} \in \mathsf{VALID} \subset \mathbb{Z}^d$ s.t. $\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod q$.

Question: Which properties of VALID do we need?

An answer: There exists a finite set $\mathcal{S}$ s.t. we can associate every $\pi \in \mathcal{S}$ with a permutation T_π of d elements, satisfying:

1. $\mathbf{w} \in \mathsf{VALID} \iff T_\pi(\mathbf{w}) \in \mathsf{VALID}$.
2. $\mathbf{w} \in \mathsf{VALID}$ and $\pi \xleftarrow{\$} \mathcal{S}$, then $T_\pi(\mathbf{w})$ is uniform in VALID.

Note: Stern's protocol corresponds to the special case when

$$\mathsf{VALID} = \mathcal{B}, \quad \mathcal{S} = \mathcal{S}_d, \quad T_\pi(\mathbf{w}) = \pi(\mathbf{w}).$$

Abstracting Stern's Protocol

Suppose we want to use Stern to prove $\mathbf{w} \in \mathsf{VALID} \subset \mathbb{Z}^d$ s.t. $\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod q$.

Question: Which properties of VALID do we need?

An answer: There exists a finite set $\mathcal{S}$ s.t. we can associate every $\pi \in \mathcal{S}$ with a permutation T_π of d elements, satisfying:

1. $\mathbf{w} \in \mathsf{VALID} \Longleftrightarrow T_\pi(\mathbf{w}) \in \mathsf{VALID}$.
2. $\mathbf{w} \in \mathsf{VALID}$ and $\pi \xleftarrow{\$} \mathcal{S}$, then $T_\pi(\mathbf{w})$ is uniform in VALID.

Note: Stern's protocol corresponds to the special case when

$$\mathsf{VALID} = \mathcal{B}, \quad \mathcal{S} = \mathcal{S}_d, \quad T_\pi(\mathbf{w}) = \pi(\mathbf{w}).$$

How does it work?

- To prove $\mathbf{w} \in \mathsf{VALID}$: sample $\pi \xleftarrow{\$} \mathcal{S}$, show that $T_\pi(\mathbf{w}) \in \mathsf{VALID}$.
- To prove $\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod q$, use usual masking vector $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_q^d$.

Outline

1. Introduction
   - Zero-Knowledge Protocols
   - Zero-Knowledge Protocols in Lattice-Based Cryptography
2. Stern-like Protocols
   - Stern-KTX Protocol
   - Abstracting Stern's Protocol
   - Techniques and Applications
3. Conclusion

Extensions

Example 1: Proving $\mathbf{x} \in \{0,1\}^m$ s.t. $\mathbf{A} \cdot \mathbf{x} = \mathbf{v} \bmod q$, for $(\mathbf{A}, \mathbf{v}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^n$.

Extensions

Example 1: Proving $\mathbf{x} \in \{0,1\}^m$ s.t. $\mathbf{A} \cdot \mathbf{x} = \mathbf{v} \bmod q$, for $(\mathbf{A}, \mathbf{v}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^n$.

Idea: $\mathbf{x}$ does not have fixed Hamming weight, so, let's make it fixed!

- Appending "dummy" entries $\{0,1\}$ to $\mathbf{x}$ to get $\mathbf{w} = \begin{pmatrix} \mathbf{x} \\ \vdots \end{pmatrix} \in \mathsf{B}_m^2$, where

$$\mathsf{B}_m^2 = \{\mathbf{w} \in \{0,1\}^{2m} : wt(\mathbf{w}) = m\}.$$

- Note that $\mathbf{x} = \left[\, \mathbf{I}_m \mid \mathbf{0}^{m \times m} \, \right] \cdot \mathbf{w}$, and let $\mathbf{M} = \mathbf{A} \cdot \left[\, \mathbf{I}_m \mid \mathbf{0}^{m \times m} \, \right] \in \mathbb{Z}_q^{n \times 2m}$. We then have $\mathbf{M} \cdot \mathbf{w} = \mathbf{A} \cdot \mathbf{x} = \mathbf{v} \bmod q$.

Now, we have an instance of the abstraction, where $d = 2m$, $\mathcal{S} = \mathcal{S}_d$, and $T_\pi(\mathbf{w}) = \pi(\mathbf{w})$.

Example 2: Proving $\mathbf{x} \in \{-1, 0, 1\}^m$ s.t. $\mathbf{A} \cdot \mathbf{x} = \mathbf{v} \bmod q$.

Extensions

Example 2: Proving $\mathbf{x} \in \{-1,0,1\}^m$ s.t. $\mathbf{A} \cdot \mathbf{x} = \mathbf{v} \bmod q$.

Idea: The coordinates of $\mathbf{x}$ are not balanced, let's make them balanced then.

- Appending "dummy" entries $\{-1,0,1\}$ to $\mathbf{x}$ to get $\mathbf{w} = \begin{pmatrix} \mathbf{x} \\ \vdots \end{pmatrix} \in \mathrm{B}_m^3$,

  where B_m^3 is the set of all vectors in $\{-1,0,1\}^{3m}$, that have exactly m coordinates -1; m coordinates 0; and m coordinates 1.
- Note that $\mathbf{x} = \left[\, \mathbf{I}_m \mid \mathbf{0}^{m\times 2m} \,\right] \cdot \mathbf{w}$, and let $\mathbf{M} = \mathbf{A} \cdot \left[\, \mathbf{I}_m \mid \mathbf{0}^{m\times 2m} \,\right] \in \mathbb{Z}_q^{n\times 3m}$. We then have $\mathbf{M} \cdot \mathbf{w} = \mathbf{A} \cdot \mathbf{x} = \mathbf{v} \bmod q$.

Again, we have an instance of the abstraction.

Decompositions

Decomposition sequence

$\forall\, \beta \in \mathbb{Z}_+$, let $\delta_\beta := \lfloor \log_2 \beta \rfloor + 1$; define $\beta_1, \ldots, \beta_{\delta_\beta}$, where $\beta_j = \left\lfloor \frac{\beta + 2^{j-1}}{2^j} \right\rfloor, \forall j$.

Property: $z \in [0, \beta] \iff \exists\, c_1, \ldots, c_{\delta_\beta} \in \{0,1\} : z = \sum_{j=1}^{\delta_\beta} \beta_j \cdot c_j$.

Decompositions

Decomposition sequence

$\forall\, \beta \in \mathbb{Z}_+$, let $\delta_\beta := \lfloor \log_2 \beta \rfloor + 1$; define $\beta_1, \ldots, \beta_{\delta_\beta}$, where $\beta_j = \left\lfloor \frac{\beta + 2^{j-1}}{2^j} \right\rfloor, \forall j$.

$$\text{Property: } z \in [0, \beta] \iff \exists\, c_1, \ldots, c_{\delta_\beta} \in \{0,1\} : z = \sum_{j=1}^{\delta_\beta} \beta_j \cdot c_j.$$

Decomposition matrix

For $m, \beta \in \mathbb{Z}_+$, define

$$\mathbf{H}_{m,\beta} := \begin{bmatrix} \beta_1 \ldots \beta_{\delta_\beta} & & \\ & \ddots & \\ & & \beta_1 \ldots \beta_{\delta_\beta} \end{bmatrix} \in \mathbb{Z}^{m \times m\delta_\beta}.$$

As a result, we have

$$\mathbf{x} \in [-\beta, \beta]^m \iff \exists\, \mathbf{x}' \in \{-1,0,1\}^{m\delta_\beta} : \mathbf{x} = \mathbf{H}_{m,\beta} \cdot \mathbf{x}'.$$

Decomposition-Extension

Example 3 The ISIS relation: $\mathbf{x} \in [-\beta, \beta]^m$ s.t. $\mathbf{A} \cdot \mathbf{x} = \mathbf{v} \bmod q$.

1. Decompose $\mathbf{x}$ into $\mathbf{x}' \in \{-1, 0, 1\}^{m\delta_\beta}$.
2. Let $\mathbf{A}' = \mathbf{A} \cdot \mathbf{H}_{m,\beta}$, then we have $\mathbf{A}' \cdot \mathbf{x}' = \mathbf{v} \bmod q$.
3. Reduce to **Example 2**.

Applications: Proving knowledge of a lattice-based signature (e.g., [GPV'08], [Boy'10], [CHKP'10]) on a publicly given message.

Decomposition-Extension

Example 3 The ISIS relation: $\mathbf{x} \in [-\beta, \beta]^m$ s.t. $\mathbf{A} \cdot \mathbf{x} = \mathbf{v} \bmod q$.

1. Decompose $\mathbf{x}$ into $\mathbf{x}' \in \{-1, 0, 1\}^{m\delta_\beta}$.
2. Let $\mathbf{A}' = \mathbf{A} \cdot \mathbf{H}_{m,\beta}$, then we have $\mathbf{A}' \cdot \mathbf{x}' = \mathbf{v} \bmod q$.
3. Reduce to **Example 2**.

Applications: Proving knowledge of a lattice-based signature (e.g., [GPV'08], [Boy'10], [CHKP'10]) on a publicly given message.

Example 4 The LWE relation (HNF): $\mathbf{s} \in [-\beta, \beta]^n$, $\mathbf{e} \in [-\beta, \beta]^m$ s.t.

$$\mathbf{A}^\top \cdot \mathbf{s} + \mathbf{e} = \mathbf{b} \bmod q.$$

1. Note that
$$\begin{bmatrix} \mathbf{A}^T & \mathbf{I}_m \end{bmatrix} \cdot \begin{pmatrix} \mathbf{s} \\ \mathbf{e} \end{pmatrix} = \mathbf{b} \bmod q.$$
2. Reduce to **Example 3**.

Applications: Proof that a given ciphertext generated of an LWE-based encryption scheme (e.g., Regev [R'05], dual-Regev [GPV'08]) is well-formed.

Advanced application 1: Group signatures

Group signatures [CH'91]:

- Allow members to anonymously sign messages on behalf of the group.
- There is a tracing authority who can identify the signer.

Advanced application 1: Group signatures

Group signatures [CH'91]:

- Allow members to anonymously sign messages on behalf of the group.
- There is a tracing authority who can identify the signer.

Common approach: "**sign-then-encrypt-then-prove**"

- Each user has a signature σ on his identity μ, certified by the manager.
- In the process of generating GS, the user encrypts μ to $\mathbf{c}$, then prove that:
  1. **He has a secret valid pair (μ, σ) w.r.t. the group public key.**
  2. $\mathbf{c}$ is a well-formed ciphertext of μ.

Advanced application 1: Group signatures

Group signatures [CH'91]:

- Allow members to anonymously sign messages on behalf of the group.
- There is a tracing authority who can identify the signer.

Common approach: "**sign-then-encrypt-then-prove**"

- Each user has a signature σ on his identity μ, certified by the manager.
- In the process of generating GS, the user encrypts μ to $\mathbf{c}$, then prove that:
  1. **He has a secret valid pair** (μ, σ) **w.r.t. the group public key.**
  2. $\mathbf{c}$ is a well-formed ciphertext of μ.

Desired building block: Zero-knowledge proof of knowledge of a valid message-signature pair for a lattice-based standard model signature.

Let's see how to do it with Boyen's signature [Boyen'10].

Example 5: Proving knowledge of a valid pair (μ, σ) for [Boyen'10].

Namely, $\mu = (\mu[1], \ldots, \mu[\ell])^T \in \{0,1\}^\ell$ and $\sigma = (\mathbf{x}_1^T \| \mathbf{x}_2^T)^T \in [-\beta, \beta]^{2m}$ s.t.

$$\mathbf{A} \cdot \mathbf{x}_1 + \mathbf{A}_0 \cdot \mathbf{x}_2 + \sum_{i=1}^{\ell} \mathbf{A}_i \cdot (\mu[i] \cdot \mathbf{x}_2) = \mathbf{u} \bmod q.$$

Example 5: Proving knowledge of a valid pair (μ, σ) for [Boyen'10].

Namely, $\mu = (\mu[1], \ldots, \mu[\ell])^T \in \{0,1\}^\ell$ and $\sigma = (\mathbf{x}_1^T \| \mathbf{x}_2^T)^T \in [-\beta, \beta]^{2m}$ s.t.

$$\mathbf{A} \cdot \mathbf{x}_1 + \mathbf{A}_0 \cdot \mathbf{x}_2 + \sum_{i=1}^{\ell} \mathbf{A}_i \cdot (\mu[i] \cdot \mathbf{x}_2) = \mathbf{u} \bmod q.$$

- Apply decomposition-extension to $\mathbf{x}_1, \mathbf{x}_2 \in [-\beta, \beta]^m$ to obtain $\mathbf{y}_1, \mathbf{y}_2 \in \mathsf{B}^3_{m\delta_\beta}$.

Example 5: Proving knowledge of a valid pair (μ, σ) for [Boyen'10].

Namely, $\mu = (\mu[1], \ldots, \mu[\ell])^T \in \{0,1\}^\ell$ and $\sigma = (\mathbf{x}_1^T \| \mathbf{x}_2^T)^T \in [-\beta, \beta]^{2m}$ s.t.

$$\mathbf{A} \cdot \mathbf{x}_1 + \mathbf{A}_0 \cdot \mathbf{x}_2 + \sum_{i=1}^{\ell} \mathbf{A}_i \cdot (\mu[i] \cdot \mathbf{x}_2) = \mathbf{u} \bmod q.$$

- Apply decomposition-extension to $\mathbf{x}_1, \mathbf{x}_2 \in [-\beta, \beta]^m$ to obtain $\mathbf{y}_1, \mathbf{y}_2 \in \mathsf{B}^3_{m\delta_\beta}$.
- Extend μ to $\mu' = (\mu[1], \ldots, \mu[\ell], \mu[\ell+1], \ldots, \mu[2\ell])^T \in \mathsf{B}^2_\ell$.

Example 5: Proving knowledge of a valid pair (μ, σ) for [Boyen'10].

Namely, $\mu = (\mu[1], \ldots, \mu[\ell])^T \in \{0,1\}^\ell$ and $\sigma = (\mathbf{x}_1^T \| \mathbf{x}_2^T)^T \in [-\beta, \beta]^{2m}$ s.t.

$$\mathbf{A} \cdot \mathbf{x}_1 + \mathbf{A}_0 \cdot \mathbf{x}_2 + \sum_{i=1}^{\ell} \mathbf{A}_i \cdot (\mu[i] \cdot \mathbf{x}_2) = \mathbf{u} \bmod q.$$

- Apply decomposition-extension to $\mathbf{x}_1, \mathbf{x}_2 \in [-\beta, \beta]^m$ to obtain $\mathbf{y}_1, \mathbf{y}_2 \in \mathsf{B}^3_{m\delta_\beta}$.
- Extend μ to $\mu' = (\mu[1], \ldots, \mu[\ell], \mu[\ell+1], \ldots, \mu[2\ell])^T \in \mathsf{B}^2_\ell$.
- Let $d = (2\ell+2)3m\delta_\beta$, form vector $\mathbf{w}$ as:

$$(\mathbf{y}_1^T \| \mathbf{y}_2^T \| \mu[1] \cdot \mathbf{y}_2^T \| \ldots \| \mu[2\ell] \cdot \mathbf{y}_2^T)^T \in \{-1, 0, 1\}^d. \quad (1)$$

Example 5: Proving knowledge of a valid pair (μ, σ) for [Boyen'10].

Namely, $\mu = (\mu[1], \ldots, \mu[\ell])^T \in \{0,1\}^\ell$ and $\sigma = (\mathbf{x}_1^T \| \mathbf{x}_2^T)^T \in [-\beta, \beta]^{2m}$ s.t.

$$\mathbf{A} \cdot \mathbf{x}_1 + \mathbf{A}_0 \cdot \mathbf{x}_2 + \sum_{i=1}^{\ell} \mathbf{A}_i \cdot (\mu[i] \cdot \mathbf{x}_2) = \mathbf{u} \bmod q.$$

- Apply decomposition-extension to $\mathbf{x}_1, \mathbf{x}_2 \in [-\beta, \beta]^m$ to obtain $\mathbf{y}_1, \mathbf{y}_2 \in \mathrm{B}^3_{m\delta_\beta}$.
- Extend μ to $\mu' = (\mu[1], \ldots, \mu[\ell], \mu[\ell+1], \ldots, \mu[2\ell])^T \in \mathrm{B}^2_\ell$.
- Let $d = (2\ell+2)3m\delta_\beta$, form vector $\mathbf{w}$ as:

$$(\mathbf{y}_1^T \| \mathbf{y}_2^T \| \mu[1] \cdot \mathbf{y}_2^T \| \ldots \| \mu[2\ell] \cdot \mathbf{y}_2^T)^T \in \{-1, 0, 1\}^d. \quad (1)$$

- We obtain $\mathbf{M} \cdot \mathbf{w} = \mathbf{u} \bmod q$, where $(\mathbf{M}, \mathbf{u})$ is public.

Example 5: Proving knowledge of a valid pair (μ, σ) for [Boyen'10].

Namely, $\mu = (\mu[1], \ldots, \mu[\ell])^T \in \{0,1\}^\ell$ and $\sigma = (\mathbf{x}_1^T \| \mathbf{x}_2^T)^T \in [-\beta, \beta]^{2m}$ s.t.

$$\mathbf{A} \cdot \mathbf{x}_1 + \mathbf{A}_0 \cdot \mathbf{x}_2 + \sum_{i=1}^{\ell} \mathbf{A}_i \cdot (\mu[i] \cdot \mathbf{x}_2) = \mathbf{u} \bmod q.$$

- Apply decomposition-extension to $\mathbf{x}_1, \mathbf{x}_2 \in [-\beta, \beta]^m$ to obtain $\mathbf{y}_1, \mathbf{y}_2 \in \mathrm{B}^3_{m\delta_\beta}$.
- Extend μ to $\mu' = (\mu[1], \ldots, \mu[\ell], \mu[\ell+1], \ldots, \mu[2\ell])^T \in \mathrm{B}^2_\ell$.
- Let $d = (2\ell+2)3m\delta_\beta$, form vector $\mathbf{w}$ as:

$$(\mathbf{y}_1^T \| \mathbf{y}_2^T \| \mu[1] \cdot \mathbf{y}_2^T \| \ldots \| \mu[2\ell] \cdot \mathbf{y}_2^T)^T \in \{-1, 0, 1\}^d. \quad (1)$$

- We obtain $\mathbf{M} \cdot \mathbf{w} = \mathbf{u} \bmod q$, where $(\mathbf{M}, \mathbf{u})$ is public.

VALID: the set of vectors of the form (1), for some $\mathbf{y}_1, \mathbf{y}_2 \in \mathrm{B}^3_{m\delta_\beta}$ and $\mu' \in \mathrm{B}^2_\ell$.

Let $\mathcal{S} = \mathcal{S}_{3m\delta_\beta} \times \mathcal{S}_{3m\delta_\beta} \times \mathcal{S}_{2\ell}$. For each $\pi = (\phi, \psi, \rho) \in \mathcal{S}$, let T_π be the permutation that transforms vector $\mathbf{t} \in (\mathbf{t}_{-1}^T \| \mathbf{t}_0^T \| \mathbf{t}_1^T \| \ldots \| \mathbf{t}_{2\ell}^T)^T \in \mathbb{Z}^D$ to:

$$T_\pi(\mathbf{t}) = \left(\phi(\mathbf{t}_{-1})^T \| \psi(\mathbf{t}_0)^T \| \psi(\mathbf{t}_{\rho(1)})^T \| \ldots \| \psi(\mathbf{t}_{\rho(2\ell)})^T\right)^T.$$

Now, we have an instance of the abstract protocol.

Advanced application 1: Group signatures

Using the proof of a message-signature pair (μ, σ) as a building block, we can obtain a group signature.

- Encrypt μ using dual-Regev [GPV'08]:

$$\mathbf{c} = \begin{bmatrix} \mathbf{B} \\ \mathbf{P} \end{bmatrix} \cdot \mathbf{s} + \begin{pmatrix} \mathbf{I}_m & \\ & \mathbf{I}_\ell \end{pmatrix} \cdot \mathbf{e} + \begin{bmatrix} \mathbf{0} \\ \lfloor \frac{q}{2} \rfloor \mathbf{I}_\ell \end{bmatrix} \cdot \mu \quad \in \mathbb{Z}_q^{m+\ell}.$$

Advanced application 1: Group signatures

Using the proof of a message-signature pair (μ, σ) as a building block, we can obtain a group signature.

- Encrypt μ using dual-Regev [GPV'08]:

$$\mathbf{c} = \begin{bmatrix} \mathbf{B} \\ \mathbf{P} \end{bmatrix} \cdot \mathbf{s} + \begin{pmatrix} \mathbf{I}_m & \\ & \mathbf{I}_\ell \end{pmatrix} \cdot \mathbf{e} + \begin{bmatrix} \mathbf{0} \\ \lfloor \frac{q}{2} \rfloor \mathbf{I}_\ell \end{bmatrix} \cdot \mu \quad \in \mathbb{Z}_q^{m+\ell}.$$

- Extend μ to $\mu' \in \mathsf{B}_{2\ell}$ as before, then use the *same* permutation τ to prove that the same μ are involved in both layers.

Advanced application 1: Group signatures

Using the proof of a message-signature pair (μ, σ) as a building block, we can obtain a group signature.

- Encrypt μ using dual-Regev [GPV'08]:

$$\mathbf{c} = \begin{bmatrix} \mathbf{B} \\ \mathbf{P} \end{bmatrix} \cdot \mathbf{s} + \begin{pmatrix} \mathbf{I}_m & \\ & \mathbf{I}_\ell \end{pmatrix} \cdot \mathbf{e} + \begin{bmatrix} \mathbf{0} \\ \lfloor \frac{q}{2} \rfloor \mathbf{I}_\ell \end{bmatrix} \cdot \mu \quad \in \mathbb{Z}_q^{m+\ell}.$$

- Extend μ to $\mu' \in \mathsf{B}_{2\ell}$ as before, then use the *same* permutation τ to prove that the same μ are involved in both layers.
- Convert the whole interactive proof into a group signature using [FS'86].

Advanced application 2: Group encryption

Group encryption [KTY'07]: dual primitive of group signature.

- Protect anonymity of ciphertext receivers who are certified group members.
- There is a tracing authority who can break anonymity should the needs arise.

Advanced application 2: Group encryption

Group encryption [KTY'07]: dual primitive of group signature.

- Protect anonymity of ciphertext receivers who are certified group members.
- There is a tracing authority who can break anonymity should the needs arise.

Common approach:

- Each member has a key pair (sk, pk) for an anonymous encryption scheme.
- Manager signs member's public key pk, and publishes (pk, σ).

Advanced application 2: Group encryption

Group encryption [KTY'07]: dual primitive of group signature.

- Protect anonymity of ciphertext receivers who are certified group members.
- There is a tracing authority who can break anonymity should the needs arise.

Common approach:

- Each member has a key pair (sk, pk) for an anonymous encryption scheme.
- Manager signs member's public key pk, and publishes (pk, σ).
- Sender encrypts a message under pk to $\mathbf{c}_{\mathsf{R}}$, also encrypts pk under the tracing authority's public key to $\mathbf{c}_{TA}$. Then proves that:

1. **$\mathbf{c}_{\mathsf{R}}$ is an encryption of some message under a hidden pk.**
2. Sender knows a sig. σ on pk and $\mathbf{c}_{\mathsf{TA}}$ is a correct encryption of that pk.

Advanced application 2: Group encryption

Group encryption [KTY'07]: dual primitive of group signature.

- Protect anonymity of ciphertext receivers who are certified group members.
- There is a tracing authority who can break anonymity should the needs arise.

Common approach:

- Each member has a key pair (sk, pk) for an anonymous encryption scheme.
- Manager signs member's public key pk, and publishes (pk, σ).
- Sender encrypts a message under pk to $\mathbf{c}_{\mathrm{R}}$, also encrypts pk under the tracing authority's public key to $\mathbf{c}_{TA}$. Then proves that:

1. **$\mathbf{c}_{\mathrm{R}}$ is an encryption of some message under a hidden pk.**
2. Sender knows a sig. σ on pk and $\mathbf{c}_{\mathrm{TA}}$ is a correct encryption of that pk.

To instantiate a GE scheme with LWE-based encryption, we will have to handle an LWE relation with hidden-but-certified matrix:

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

We call this **"quadratic relation"**.

Dealing with Quadratic Relations

Example 6: Given $\mathbf{b} \in \mathbb{Z}_q^m$, prove that $\mathbf{b} = \mathbf{X} \cdot \mathbf{s} + \mathbf{e} \bmod q$, where $\mathbf{X} \in \mathbb{Z}_q^{m \times n}$, $\mathbf{s} \in \mathbb{Z}_q^n$ satisfy additional relations.

Dealing with Quadratic Relations

Example 6: Given $\mathbf{b} \in \mathbb{Z}_q^m$, prove that $\mathbf{b} = \mathbf{X} \cdot \mathbf{s} + \mathbf{e} \bmod q$, where $\mathbf{X} \in \mathbb{Z}_q^{m \times n}$, $\mathbf{s} \in \mathbb{Z}_q^n$ satisfy additional relations.

First step: Transforming $\mathbf{X} \cdot \mathbf{s} = (\text{public matrix}) \cdot (\text{secret vector}) \bmod q$.

Dealing with Quadratic Relations

Example 6: Given $\mathbf{b} \in \mathbb{Z}_q^m$, prove that $\mathbf{b} = \mathbf{X} \cdot \mathbf{s} + \mathbf{e} \bmod q$, where $\mathbf{X} \in \mathbb{Z}_q^{m \times n}$, $\mathbf{s} \in \mathbb{Z}_q^n$ satisfy additional relations.

First step: Transforming $\mathbf{X} \cdot \mathbf{s} = \text{(public matrix)} \cdot \text{(secret vector)} \bmod q$.

Let $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{Z}_q^m$ be columns of $\mathbf{X}$, and $s_1, \ldots, s_n \in \mathbb{Z}_q$ be the entries of $\mathbf{s}$.

Note that:

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

Dealing with Quadratic Relations

Example 6: Given $\mathbf{b} \in \mathbb{Z}_q^m$, prove that $\mathbf{b} = \mathbf{X} \cdot \mathbf{s} + \mathbf{e} \bmod q$, where $\mathbf{X} \in \mathbb{Z}_q^{m \times n}$, $\mathbf{s} \in \mathbb{Z}_q^n$ satisfy additional relations.

First step: Transforming $\mathbf{X} \cdot \mathbf{s} = \text{(public matrix)} \cdot \text{(secret vector)} \bmod q$.
Let $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{Z}_q^m$ be columns of $\mathbf{X}$, and $s_1, \ldots, s_n \in \mathbb{Z}_q$ be the entries of $\mathbf{s}$.
Note that:

1. $\mathbf{X} \cdot \mathrm{s} = \sum_{i=1}^n \mathbf{x}_i \cdot s_i$.
2. $\mathbf{x}_i \cdot s_i = \mathbf{H}_{m,q-1} \cdot \left(x_{i,1} \cdot s_i, \ldots x_{i,mk} \cdot s_i\right)^T$, where $k = \lfloor \log_2(q-1) \rfloor + 1$.

Dealing with Quadratic Relations

Example 6: Given $\mathbf{b} \in \mathbb{Z}_q^m$, prove that $\mathbf{b} = \mathbf{X} \cdot \mathbf{s} + \mathbf{e} \bmod q$, where $\mathbf{X} \in \mathbb{Z}_q^{m \times n}$, $\mathbf{s} \in \mathbb{Z}_q^n$ satisfy additional relations.

First step: Transforming $\mathbf{X} \cdot \mathbf{s} = \text{(public matrix)} \cdot \text{(secret vector)} \bmod q$.

Let $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{Z}_q^m$ be columns of $\mathbf{X}$, and $s_1, \ldots, s_n \in \mathbb{Z}_q$ be the entries of $\mathbf{s}$.

Note that:

1. $\mathbf{X} \cdot \mathrm{s} = \sum_{i=1}^{n} \mathbf{x}_i \cdot s_i$.
2. $\mathbf{x}_i \cdot s_i = \mathbf{H}_{m,q-1} \cdot \big(x_{i,1} \cdot s_i, \ldots x_{i,mk} \cdot s_i\big)^T$, where $k = \lfloor \log_2(q-1) \rfloor + 1$.
3. $x_{i,j} \cdot s_i = x_{i,j} \cdot (q_1, \ldots, q_k) \cdot (s_{i,1}, \ldots, s_{i,k})^T = (q_1, \ldots, q_k) \cdot (x_{i,j} \cdot s_{i,1}, \ldots, x_{i,j} \cdot s_{i,k})^T$

Dealing with Quadratic Relations

Example 6: Given $\mathbf{b} \in \mathbb{Z}_q^m$, prove that $\mathbf{b} = \mathbf{X} \cdot \mathbf{s} + \mathbf{e} \bmod q$, where $\mathbf{X} \in \mathbb{Z}_q^{m \times n}$, $\mathbf{s} \in \mathbb{Z}_q^n$ satisfy additional relations.

First step: Transforming $\mathbf{X} \cdot \mathbf{s} = (\text{public matrix}) \cdot (\text{secret vector}) \bmod q$.
Let $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{Z}_q^m$ be columns of $\mathbf{X}$, and $s_1, \ldots, s_n \in \mathbb{Z}_q$ be the entries of $\mathbf{s}$.
Note that:

1. $\mathbf{X} \cdot \mathrm{s} = \sum_{i=1}^{n} \mathbf{x}_i \cdot s_i$.
2. $\mathbf{x}_i \cdot s_i = \mathbf{H}_{m,q-1} \cdot \left(x_{i,1} \cdot s_i, \ldots x_{i,mk} \cdot s_i\right)^T$, where $k = \lfloor \log_2(q-1) \rfloor + 1$.
3. $x_{i,j} \cdot s_i = x_{i,j} \cdot (q_1, \ldots, q_k) \cdot (s_{i,1}, \ldots, s_{i,k})^T = (q_1, \ldots, q_k) \cdot (x_{i,j} \cdot s_{i,1}, \ldots, x_{i,j} \cdot s_{i,k})^T$

$x_{i,j} \cdot s_i$ has form $(\text{public matrix}) \cdot (\text{secret vector})$ $\rightarrow$ so does $\mathbf{x}_i \cdot s_i$ $\rightarrow$ so does $\mathbf{X} \cdot \mathbf{s}$:

$$\mathbf{X} \cdot \mathbf{s} = \mathbf{Q} \cdot \mathbf{z} \bmod q,$$

where $\mathbf{Q} \in \mathbb{Z}_q^{m \times nmk^2}$ and $\mathbf{z} \in \{0,1\}^{nmk^2}$.

Dealing with Quadratic Relations

Example 6: Given $\mathbf{b} \in \mathbb{Z}_q^m$, prove that $\mathbf{b} = \mathbf{X} \cdot \mathbf{s} + \mathbf{e} \bmod q$, where $\mathbf{X} \in \mathbb{Z}_q^{m \times n}$, $\mathbf{s} \in \mathbb{Z}_q^n$ satisfy additional relations.

First step: Transforming $\mathbf{X} \cdot \mathbf{s} = (\text{public matrix}) \cdot (\text{secret vector}) \bmod q$.
Let $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{Z}_q^m$ be columns of $\mathbf{X}$, and $s_1, \ldots, s_n \in \mathbb{Z}_q$ be the entries of $\mathbf{s}$.
Note that:

1. $\mathbf{X} \cdot \mathrm{s} = \sum_{i=1}^n \mathbf{x}_i \cdot s_i$.
2. $\mathbf{x}_i \cdot s_i = \mathbf{H}_{m,q-1} \cdot \left(x_{i,1} \cdot s_i, \ldots x_{i,mk} \cdot s_i\right)^T$, where $k = \lfloor \log_2(q-1) \rfloor + 1$.
3. $x_{i,j} \cdot s_i = x_{i,j} \cdot (q_1, \ldots, q_k) \cdot (s_{i,1}, \ldots, s_{i,k})^T = (q_1, \ldots, q_k) \cdot (x_{i,j} \cdot s_{i,1}, \ldots, x_{i,j} \cdot s_{i,k})^T$

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

$$\mathbf{X} \cdot \mathbf{s} = \mathbf{Q} \cdot \mathbf{z} \bmod q,$$

where $\mathbf{Q} \in \mathbb{Z}_q^{m \times nmk^2}$ and $\mathbf{z} \in \{0,1\}^{nmk^2}$. But...the harder part is still ahead!

Vector $\mathbf{z}$ still has a quadratic nature: each of its entries is a product of a bit from $\mathbf{X}$ and a bit from $\mathbf{s}$. And these component bits must also satisfy other relations!

Dealing with Quadratic Relations

Divide-and-conquer: Let us view the problem as a bunch of sub-problems: Proving that z has the form $c_1 \cdot c_2$, while "keeping track" of the bits c_1 and c_2.

Dealing with Quadratic Relations

Divide-and-conquer: Let us view the problem as a bunch of sub-problems: Proving that z has the form $c_1 \cdot c_2$, while "keeping track" of the bits c_1 and c_2.

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

$$\mathrm{ext}(c_1, c_2) = (\overline{c}_1 \cdot \overline{c}_2, \overline{c}_1 \cdot c_2, c_1 \cdot \overline{c}_2, c_1 \cdot c_2)^\top \in \{0,1\}^4.$$

- For $b_1, b_2 \in \{0,1\}$, define the permutation T_{b_1,b_2} that transforms vector $\mathbf{v} = (v_{0,0}, v_{0,1}, v_{1,0}, v_{1,1})^\top \in \mathbb{Z}^4$ to vector $(v_{b_1,b_2}, v_{b_1,\overline{b}_2}, v_{\overline{b}_1,b_2}, v_{\overline{b}_1,\overline{b}_2})^\top$.
  Note that, for all $c_1, c_2, b_1, b_2 \in \{0,1\}$, we have the following:

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

Dealing with Quadratic Relations

Divide-and-conquer: Let us view the problem as a bunch of sub-problems: Proving that z has the form $c_1 \cdot c_2$, while "keeping track" of the bits c_1 and c_2.

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

$$\mathrm{ext}(c_1, c_2) = (\overline{c}_1 \cdot \overline{c}_2, \overline{c}_1 \cdot c_2, c_1 \cdot \overline{c}_2, c_1 \cdot c_2)^\top \in \{0,1\}^4.$$

- For $b_1, b_2 \in \{0,1\}$, define the permutation T_{b_1,b_2} that transforms vector $\mathbf{v} = (v_{0,0}, v_{0,1}, v_{1,0}, v_{1,1})^\top \in \mathbb{Z}^4$ to vector $(v_{b_1,b_2}, v_{b_1,\overline{b}_2}, v_{\overline{b}_1,b_2}, v_{\overline{b}_1,\overline{b}_2})^\top$.
  Note that, for all $c_1, c_2, b_1, b_2 \in \{0,1\}$, we have the following:

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

Example: $c_1 = 1, c_2 = 0$. Then $\mathrm{ext}(c_1, c_2) = (0 \cdot 1, 0 \cdot 0, 1 \cdot 1, 1 \cdot 0)^T = (0,0,1,0)^T$. Then we have $v_{0,0} = 0, v_{0,1} = 0, v_{1,0} = 1, v_{1,1} = 0$. Now, let $b_1 = 1, b_2 = 1$.

$$T_{1,1}(\mathrm{ext}(1,0)) = (0,1,0,0)^T = \mathrm{ext}(0,1) = \mathrm{ext}(1 \oplus 1, 0 \oplus 1).$$

Dealing with Quadratic Relations

Divide-and-conquer: Let us view the problem as a bunch of sub-problems: Proving that z has the form $c_1 \cdot c_2$, while "keeping track" of the bits c_1 and c_2.

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

$$\mathrm{ext}(c_1, c_2) = (\overline{c}_1 \cdot \overline{c}_2, \overline{c}_1 \cdot c_2, c_1 \cdot \overline{c}_2, c_1 \cdot c_2)^\top \in \{0,1\}^4.$$

- For $b_1, b_2 \in \{0,1\}$, define the permutation T_{b_1,b_2} that transforms vector $\mathbf{v} = (v_{0,0}, v_{0,1}, v_{1,0}, v_{1,1})^\top \in \mathbb{Z}^4$ to vector $(v_{b_1,b_2}, v_{b_1,\overline{b}_2}, v_{\overline{b}_1,b_2}, v_{\overline{b}_1,\overline{b}_2})^\top$.
  Note that, for all $c_1, c_2, b_1, b_2 \in \{0,1\}$, we have the following:

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

Example: $c_1 = 1, c_2 = 0$. Then $\mathrm{ext}(c_1, c_2) = (0 \cdot 1, 0 \cdot 0, 1 \cdot 1, 1 \cdot 0)^T = (0,0,1,0)^T$. Then we have $v_{0,0} = 0, v_{0,1} = 0, v_{1,0} = 1, v_{1,1} = 0$. Now, let $b_1 = 1, b_2 = 1$.

$$T_{1,1}(\mathrm{ext}(1,0)) = (0,1,0,0)^T = \mathrm{ext}(0,1) = \mathrm{ext}(1 \oplus 1, 0 \oplus 1).$$

Solution to sub-problem: extend z to $\mathbf{z}$, then permute it with random bits b_1, b_2. To "keep track", use the same b_1, b_2 at other appearances of c_1, c_2, resp.

Conclusion

- Stern's protocol has been developing into a strong tool for privacy-preserving lattice-based crypto.
- 4 techniques: decomposing, extending, permuting, masking.

Thank you!