# Shorter Lattice-Based Group Signatures via "Almost Free" Encryption and Other Optimizations

Vadim Lyubashevsky[1], Ngoc Khanh Nguyen[12], Maxime Plançon[12], and Gregor Seiler[12]

[1]IBM Research Europe, Switzerland, [2]ETH Zurich

November 29, 2021

# Overview of the paper

Lattice-based group signature construction from [dPLS18]

\+ Lattice-based ZK proofs contributions from [ALS20, ENS20, LNS20, BLS19, ESLL19]

→ This paper

# BDLOP lattice-based commitment scheme

Notations : $q$ prime, $d$ power of 2, $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$, $\mathcal{R}_q = \mathbb{Z}_q/(X^d + 1)$. Vectors are lower case bold letters, matrices are upper case bold letters.

# BDLOP lattice-based commitment scheme

Notations : $q$ prime, $d$ power of 2, $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$, $\mathcal{R}_q = \mathbb{Z}_q/(X^d + 1)$. Vectors are lower case bold letters, matrices are upper case bold letters.

## Definition of BDLOP commitment scheme

**Setup :** Uniformly random matrix $\mathbf{A}_0 \in \mathcal{R}_q^{n \times k}$, uniformly random vectors $\mathbf{a}_1, \ldots, \mathbf{a}_\alpha \in \mathcal{R}_q^k$.

**Commit :** To commit to $m_1, \ldots, m_\alpha \in \mathcal{R}_q$

1. Sample $\mathbf{r} \in \mathcal{R}_q^k$ with short coefficients

# BDLOP lattice-based commitment scheme

Notations : $q$ prime, $d$ power of 2, $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$, $\mathcal{R}_q = \mathbb{Z}_q/(X^d + 1)$. Vectors are lower case bold letters, matrices are upper case bold letters.

## Definition of BDLOP commitment scheme

**Setup :** Uniformly random matrix $\mathbf{A}_0 \in \mathcal{R}_q^{n \times k}$, uniformly random vectors $\mathbf{a}_1, \ldots, \mathbf{a}_\alpha \in \mathcal{R}_q^k$.

**Commit :** To commit to $m_1, \ldots, m_\alpha \in \mathcal{R}_q$

1. Sample $\mathbf{r} \in \mathcal{R}_q^k$ with short coefficients
2. Output $\mathbf{f} = \begin{bmatrix} \mathbf{A}_0 \\ \mathbf{a}_1^T \\ \vdots \\ \mathbf{a}_\alpha^T \end{bmatrix} \mathbf{r} + \begin{bmatrix} \mathbf{0} \\ m_1 \\ \vdots \\ m_\alpha \end{bmatrix}$

# Interactive ZKPoK from BDLOP

Notations : $\mathbf{f} = (\mathbf{A}_0\mathbf{r}\ \mathbf{a}_1^T\mathbf{r} + m)$, $c \in \mathcal{C}$ is such that $c \in \{-1, 0, 1\}^d$, $D_\sigma$ is the discrete Gaussian of standard deviation $\sigma$ over $\mathbb{Z}$.

# Interactive ZKPoK from BDLOP

Notations : $\mathbf{f} = (\mathbf{A}_0\mathbf{r}\ \mathbf{a}_1^T\mathbf{r} + m)$, $c \in \mathcal{C}$ is such that $c \in \{-1, 0, 1\}^d$, $D_\sigma$ is the discrete Gaussian of standard deviation $\sigma$ over $\mathbb{Z}$.

| Prover | | Verifier |
|---|---|---|
| Sample $\mathbf{y} \leftarrow D_\sigma$ | | |
| $\mathbf{w} = \mathbf{A}_0\mathbf{y}$ | $\xrightarrow{\mathbf{f},\mathbf{w}}$ | |

# Interactive ZKPoK from BDLOP

Notations : $\mathbf{f} = (\mathbf{A}_0\mathbf{r}\ \mathbf{a}_1^T\mathbf{r} + m)$, $c \in \mathcal{C}$ is such that $c \in \{-1, 0, 1\}^d$, $D_\sigma$ is the discrete Gaussian of standard deviation $\sigma$ over $\mathbb{Z}$.

| Prover | | Verifier |
|---|---|---|
| Sample $\mathbf{y} \leftarrow D_\sigma$ | | |
| $\mathbf{w} = \mathbf{A}_0\mathbf{y}$ | $\xrightarrow{\mathbf{f},\mathbf{w}}$ | |
| | | $c \leftarrow \mathcal{C}$ |
| | $\xleftarrow{c}$ | |

# Interactive ZKPoK from BDLOP

Notations : $\mathbf{f} = (\mathbf{A}_0\mathbf{r}\ \mathbf{a}_1^T\mathbf{r} + m)$, $c \in \mathcal{C}$ is such that $c \in \{-1, 0, 1\}^d$, $D_\sigma$ is the discrete Gaussian of standard deviation $\sigma$ over $\mathbb{Z}$.

| Prover | | Verifier |
|---|---|---|
| Sample $\mathbf{y} \leftarrow D_\sigma$ | | |
| $\mathbf{w} = \mathbf{A}_0\mathbf{y}$ | $\xrightarrow{\mathbf{f},\mathbf{w}}$ | |
| | | $c \leftarrow \mathcal{C}$ |
| | $\xleftarrow{c}$ | |
| $\mathbf{z} = \mathbf{y} + c\mathbf{r}$ | | |
| Rejection sampling on $\mathbf{z}$ | $\xrightarrow{\mathbf{z}}$ | |
| | | Check $\mathbf{A}_0\mathbf{z} = \mathbf{w} + c\mathbf{r}$ |

# Efficient lattice-based ZK proofs

Some statements that can be proved efficiently from BDLOP commitments :

1. linear relations : $m_1 = \lambda m_2$ for some $\lambda \in \mathcal{R}_q$ (very cheap)

# Efficient lattice-based ZK proofs

Some statements that can be proved efficiently from BDLOP commitments :

1. linear relations : $m_1 = \lambda m_2$ for some $\lambda \in \mathcal{R}_q$ (very cheap)
2. product relations : $m_1 = m_2 m_3$ (less cheap)

# Efficient lattice-based ZK proofs

Some statements that can be proved efficiently from BDLOP commitments :

1. linear relations : $m_1 = \lambda m_2$ for some $\lambda \in \mathcal{R}_q$ (very cheap)
2. product relations : $m_1 = m_2 m_3$ (less cheap)
3. "unstructured" linear relations : $\mathbf{Q}m_1 = m_2$, where $\mathbf{Q} \in \mathbb{Z}_q^{*\times d}, m_1, m_2 \in \mathbb{Z}_q^d$ (less cheap)

# What is a group signature ?

# What is a group signature ?

Entities : Setup authority, group manager (or opener), group member (or user)

# What is a group signature ?

Entities : Setup authority, group manager (or opener), group member (or user)

- The setup authority generates a public key for the group, the opener's secret key $s$, and individual signing keys $s_i$ for each identity $i \in \mathcal{I}$.
- The group manager receives $s$, the group member of identity $i$ receives the signing key $s_i$

# What is a group signature ?

Entities : Setup authority, group manager (or opener), group member (or user)

- The setup authority generates a public key for the group, the opener's secret key $s$, and individual signing keys $s_i$ for each identity $i \in \mathcal{I}$.
- The group manager receives $s$, the group member of identity $i$ receives the signing key $s_i$
- User $i$ shall be able to produce a signature $\sigma$ of a message $m$ under the public key of the group.

# What is a group signature ?

Entities : Setup authority, group manager (or opener), group member (or user)

- The setup authority generates a public key for the group, the opener's secret key $s$, and individual signing keys $s_i$ for each identity $i \in \mathcal{I}$.
- The group manager receives $s$, the group member of identity $i$ receives the signing key $s_i$
- User $i$ shall be able to produce a signature $\sigma$ of a message $m$ under the public key of the group.
- The Group Manager can open $\sigma$ to recover the identity $i$ of the signer.

# Security properties

1. **Anonymity.** The adversary who knows all the signing keys $s_i$ cannot distinguish between signatures produced by any two users.

   Intuition : a signature shall not leak the identity of the signer

# Security properties

1. **Anonymity.** The adversary who knows all the signing keys $s_i$ cannot distinguish between signatures produced by any two users.

   Intuition : a signature shall not leak the identity of the signer

2. **Traceability.** The adversary who possesses signing keys to all users in some set $S$, and the Opener's secret key, cannot create a valid signature that the Opener will decrypt to some identity $i \notin S$.

   Intuition : If $S = \emptyset$, then the adversary shall not be able to produce a valid signature.

# Previous group signature construction dPLS18

The set of identities $\mathcal{I}$ is those elements of $\mathcal{R}_q$ that are stable under 2 given automorphisms.

# Previous group signature construction dPLS18

The set of identities $\mathcal{I}$ is those elements of $\mathcal{R}_q$ that are stable under 2 given automorphisms.

## Setup algorithm

1. Generate $\mathbf{A} \leftarrow \mathcal{R}_q^{n \times k}$, short trapdoor $\mathbf{R} \in \mathcal{R}_q^{k \times 3k}$ and set $\mathbf{B} = \mathbf{AR}$.

# Previous group signature construction dPLS18

The set of identities $\mathcal{I}$ is those elements of $\mathcal{R}_q$ that are stable under 2 given automorphisms.

## Setup algorithm

1. Generate $\mathbf{A} \leftarrow \mathcal{R}_q^{n \times k}$, short trapdoor $\mathbf{R} \in \mathcal{R}_q^{k \times 3k}$ and set $\mathbf{B} = \mathbf{AR}$.
2. Sample $(\mathbf{s}_1, \mathbf{s}_2) \leftarrow D_s$ and let $\mathbf{u} = \left[\mathbf{A}|\mathbf{B}\right] \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix}$. The public key is $(\mathbf{A}, \mathbf{B}, \mathbf{u})$.

# Previous group signature construction dPLS18

The set of identities $\mathcal{I}$ is those elements of $\mathcal{R}_q$ that are stable under 2 given automorphisms.

## Setup algorithm

1. Generate $\mathbf{A} \leftarrow \mathcal{R}_q^{n \times k}$, short trapdoor $\mathbf{R} \in \mathcal{R}_q^{k \times 3k}$ and set $\mathbf{B} = \mathbf{AR}$.
2. Sample $(\mathbf{s}_1, \mathbf{s}_2) \leftarrow D_s$ and let $\mathbf{u} = \left[\mathbf{A}|\mathbf{B}\right] \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix}$. The public key is $(\mathbf{A}, \mathbf{B}, \mathbf{u})$.
3. User $i$'s signing key : Sample $(\mathbf{s}_1^i, \mathbf{s}_2^i) \leftarrow D_s$ such that (conditioned on) $\left[\mathbf{A}|\mathbf{AR} - i\mathbf{G}\right] \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix}$, where $\mathbf{G}$ is a simple matrix that allows such sampling using the trapdoor $\mathbf{R}$.

# Previous group signature construction dPLS18

The set of identities $\mathcal{I}$ is those elements of $\mathcal{R}_q$ that are stable under 2 given automorphisms.

## Setup algorithm

1. Generate $\mathbf{A} \leftarrow \mathcal{R}_q^{n \times k}$, short trapdoor $\mathbf{R} \in \mathcal{R}_q^{k \times 3k}$ and set $\mathbf{B} = \mathbf{AR}$.
2. Sample $(\mathbf{s}_1, \mathbf{s}_2) \leftarrow D_s$ and let $\mathbf{u} = \left[\mathbf{A}|\mathbf{B}\right] \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix}$. The public key is $(\mathbf{A}, \mathbf{B}, \mathbf{u})$.
3. User $i$'s signing key : Sample $(\mathbf{s}_1^i, \mathbf{s}_2^i) \leftarrow D_s$ such that (conditioned on) $\left[\mathbf{A}|\mathbf{AR} - i\mathbf{G}\right] \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix}$, where $\mathbf{G}$ is a simple matrix that allows such sampling using the trapdoor $\mathbf{R}$.
4. Generate a key pair of a verifiable encryption scheme.

# Signing

User $i$'s signature of a message $M$ :

- $(\mathbf{f}_0, f_1)$ BDLOP commitment to $i$ with randomness $\mathbf{r}$ and compute a NIZK $\pi_1$ that $i \in \mathcal{I}$ (that is $\sigma(i) = i$ for both automorphisms defining $\mathcal{I}$)

# Signing

User $i$'s signature of a message $M$ :

- $(\mathbf{f}_0, f_1)$ BDLOP commitment to $i$ with randomness $\mathbf{r}$ and compute a NIZK $\pi_1$ that $i \in \mathcal{I}$ (that is $\sigma(i) = i$ for both automorphisms defining $\mathcal{I}$)
- Set $\mathbf{v}^T = \left[\mathbf{A}|\mathbf{B} + f_1\mathbf{G}\right]^*$

# Signing

User $i$'s signature of a message $M$ :

- $(\mathbf{f}_0, f_1)$ BDLOP commitment to $i$ with randomness $\mathbf{r}$ and compute a NIZK $\pi_1$ that $i \in \mathcal{I}$ (that is $\sigma(i) = i$ for both automorphisms defining $\mathcal{I}$)
- Set $\mathbf{v}^T = \left[\mathbf{A}|\mathbf{B} + f_1\mathbf{G}\right]^*$
- Compute a NIZKP $\pi_2$ that $\left[\mathbf{A}|\mathbf{AR} - i\mathbf{G}\right] \begin{bmatrix}\mathbf{s}_1 \\ \mathbf{s}_2\end{bmatrix}^*$.

# Signing

User $i$'s signature of a message $M$ :

- $(\mathbf{f}_0, f_1)$ BDLOP commitment to $i$ with randomness $\mathbf{r}$ and compute a NIZK $\pi_1$ that $i \in \mathcal{I}$ (that is $\sigma(i) = i$ for both automorphisms defining $\mathcal{I}$)
- Set $\mathbf{v}^T = \left[\mathbf{A}|\mathbf{B} + f_1\mathbf{G}\right]^*$
- Compute a NIZKP $\pi_2$ that $\left[\mathbf{A}|\mathbf{AR} - i\mathbf{G}\right] \begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\end{bmatrix}^*$.
- Encrypt $\mathbf{r}$ with the group manager's verifiable encryption public key, and compute a NIZK $\pi_3$ that the ciphertext $\mathbf{c}$ is well-formed.

The signature is $(\mathbf{f}, \mathbf{c}, \pi_1, \pi_2, \pi_3)$. Verification is simply verifying $\pi_1, \pi_2, \pi_3$. Opening is simply decrypting $\mathbf{r}$ from $\mathbf{c}$.

# This group signature

We improve upon the previous construction on the following :

1. We extend the scheme to Module-LWE (in the previous construction, $n = 1$ and the underlying problem is Ring-LWE)

# This group signature

We improve upon the previous construction on the following :

1. We extend the scheme to Module-LWE (in the previous construction, $n = 1$ and the underlying problem is Ring-LWE)
2. Simpler and more efficient way to prove $\left[\mathbf{A}|\mathbf{AR} - i\mathbf{G}\right] \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix}$

# This group signature

We improve upon the previous construction on the following :

1. We extend the scheme to Module-LWE (in the previous construction, $n = 1$ and the underlying problem is Ring-LWE)
2. Simpler and more efficient way to prove $\left[\mathbf{A}|\mathbf{AR} - i\mathbf{G}\right] \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix}$
3. More efficient membership proof for $\pi_1 : i \in \mathcal{I}$.

# This group signature

We improve upon the previous construction on the following :

1. We extend the scheme to Module-LWE (in the previous construction, $n = 1$ and the underlying problem is Ring-LWE)
2. Simpler and more efficient way to prove $\left[\mathbf{A}|\mathbf{AR} - i\mathbf{G}\right] \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix}$
3. More efficient membership proof for $\pi_1 : i \in \mathcal{I}$.
4. Encryption almost comes for free, so does the well-formedness ZKP

Overall, we shrink the size of the signature from 580 KB to 203 KB.

# Proving knowledge of $(\mathbf{s}_1^i, \mathbf{s}_2^i)$

We need to prove knowledge of $(\mathbf{s}_1^i, \mathbf{s}_2^i)$ such that

$$\left[\mathbf{A}|\mathbf{AR} - i\mathbf{G}\right] \begin{bmatrix} \mathbf{s}_1^i \\ \mathbf{s}_2^i \end{bmatrix} = \mathbf{u},$$

where $\mathbf{A}, \mathbf{B}, \mathbf{G}, \mathbf{u}$ are public, and $i$ is committed to.

# Proving knowledge of $(\mathbf{s}_1^i, \mathbf{s}_2^i)$

We need to prove knowledge of $(\mathbf{s}_1^i, \mathbf{s}_2^i)$ such that

$$\left[\mathbf{A}|\mathbf{AR} - i\mathbf{G}\right] \begin{bmatrix} \mathbf{s}_1^i \\ \mathbf{s}_2^i \end{bmatrix} = \mathbf{u},$$

where $\mathbf{A}, \mathbf{B}, \mathbf{G}, \mathbf{u}$ are public, and $i$ is committed to.

**dPLS18 solution**

In dPLS18, this proof is done using a trick that increases the length of the Ring-SIS solution to be extracted, which affects badly the parameters.

# Proving knowledge of $(\mathbf{s}_1^i, \mathbf{s}_2^i)$

We need to prove knowledge of $(\mathbf{s}_1^i, \mathbf{s}_2^i)$ such that

$$\left[\mathbf{A}|\mathbf{AR} - i\mathbf{G}\right] \begin{bmatrix} \mathbf{s}_1^i \\ \mathbf{s}_2^i \end{bmatrix} = \mathbf{u},$$

where $\mathbf{A}, \mathbf{B}, \mathbf{G}, \mathbf{u}$ are public, and $i$ is committed to.

## dPLS18 solution

In dPLS18, this proof is done using a trick that increases the length of the Ring-SIS solution to be extracted, which affects badly the parameters.

## Can we use the product proof from BDLOP commitments ?

The equation to be proven can be written as $\mathbf{A}\mathbf{s}_1^i + \mathbf{B}\mathbf{s}_2^i - \mathbf{G}i\mathbf{s}_2^i = \mathbf{u}$. Can we do a product proof ? Yes, but committing to $\mathbf{s}_1^i$ and $\mathbf{s}_2^i$ would be a disaster for the length of the signature.

# Proving knowledge of $(\mathbf{s}_1^i, \mathbf{s}_2^i)$

$\mathbf{A}' = [\mathbf{A}|\mathbf{B} - i\mathbf{G}]$.

**Our solution**

If $\mathbf{A}'$ was public, the verification equation would be $\mathbf{A}'\mathbf{z} = \mathbf{w} + c\mathbf{u}$, where $\mathbf{w}$ would be $\mathbf{A}'\mathbf{y}$ for some Gaussian $\mathbf{y}$, $c$ a challenge and $\mathbf{z} = \mathbf{y} + c[\mathbf{s}_1^i \mathbf{s}_2^i]^T$.

# Proving knowledge of $(\mathbf{s}_1^i, \mathbf{s}_2^i)$

$\mathbf{A}' = [\mathbf{A}|\mathbf{B} - i\mathbf{G}]$.

## Our solution

If $\mathbf{A}'$ was public, the verification equation would be $\mathbf{A}'\mathbf{z} = \mathbf{w} + c\mathbf{u}$, where $\mathbf{w}$ would be $\mathbf{A}'\mathbf{y}$ for some Gaussian $\mathbf{y}$, $c$ a challenge and $\mathbf{z} = \mathbf{y} + c[\mathbf{s}_1^i \mathbf{s}_2^i]^T$.

From the homomorphic properties of the BDLOP commitment scheme, the verifier can infer a commitment to $\mathbf{A}' = [\mathbf{A}|\mathbf{B} - i\mathbf{G}]$ from the commitment to $i$.

# Proving knowledge of $(\mathbf{s}_1^i, \mathbf{s}_2^i)$

$\mathbf{A}' = [\mathbf{A}|\mathbf{B} - i\mathbf{G}]$.

**Our solution**

If $\mathbf{A}'$ was public, the verification equation would be $\mathbf{A}'\mathbf{z} = \mathbf{w} + c\mathbf{u}$, where $\mathbf{w}$ would be $\mathbf{A}'\mathbf{y}$ for some Gaussian $\mathbf{y}$, $c$ a challenge and $\mathbf{z} = \mathbf{y} + c[\mathbf{s}_1^i \mathbf{s}_2^i]^T$.

From the homomorphic properties of the BDLOP commitment scheme, the verifier can infer a commitment to $\mathbf{A}' = [\mathbf{A}|\mathbf{B} - i\mathbf{G}]$ from the commitment to $i$.

Our solution is to commit to $\mathbf{w}$ instead of sending it. The equation $\mathbf{A}'\mathbf{z} = \mathbf{w} + c\mathbf{u}$ is then linear in the committed messages, so we can give a (relatively cheap) BDLOP linear proof of the verification equation.

# Membership proof $i \in \mathcal{I}$

We chose for $\mathcal{I}$ the set $\{i' \in \mathbb{Z}_q, i' \leq 2^{d-1}\} \subset \mathcal{R}_q$.

# Membership proof $i \in \mathcal{I}$

We chose for $\mathcal{I}$ the set $\{i' \in \mathbb{Z}_q, i' \leq 2^{d-1}\} \subset \mathcal{R}_q$.

## ZK proof that $i \in \mathcal{I}$

We add an extra commitment to $i_b$, the inverse NTT of the binary representation of $i$. Only remains to prove two things :

1. $i_b$'s NTT is binary -> $\text{NTT}(i_b) \odot (\text{NTT}(i_b) - 1) = 0$ -> $\text{NTT}(i_b(i_b - 1)) = 0$ -> $i_b(i_b - 1) = 0$ is a product proof.

# Membership proof $i \in \mathcal{I}$

We chose for $\mathcal{I}$ the set $\{i' \in \mathbb{Z}_q, i' \leq 2^{d-1}\} \subset \mathcal{R}_q$.

## ZK proof that $i \in \mathcal{I}$

We add an extra commitment to $i_b$, the inverse NTT of the binary representation of $i$. Only remains to prove two things :

1. $i_b$'s NTT is binary -> $\text{NTT}(i_b) \odot (\text{NTT}(i_b) - 1) = 0$ -> $\text{NTT}(i_b(i_b - 1)) = 0$ -> $i_b(i_b - 1) = 0$ is a product proof.
2. $\mathbf{Q} \cdot \text{NTT}(i_b) = \text{NTT}(i)$, where $\mathbf{Q} = \begin{bmatrix} 1 & 2 & 4 & \dots & 2^{d-1} \\ 1 & 2 & 4 & \dots & 2^{d-1} \\ \vdots & \vdots & & & \vdots \\ 1 & 2 & 4 & \dots & 2^{d-1} \end{bmatrix}$. This means that all of $i$'s NTT coefficients are equal, hence $i$ is an integer.

# Membership proof $i \in \mathcal{I}$

We chose for $\mathcal{I}$ the set $\{i' \in \mathbb{Z}_q, i' \leq 2^{d-1}\} \subset \mathcal{R}_q$.

## ZK proof that $i \in \mathcal{I}$

We add an extra commitment to $i_b$, the inverse NTT of the binary representation of $i$. Only remains to prove two things :

1. $i_b$'s NTT is binary -> $\text{NTT}(i_b) \odot (\text{NTT}(i_b) - 1) = 0$ -> $\text{NTT}(i_b(i_b - 1)) = 0$ -> $i_b(i_b - 1) = 0$ is a product proof.
2. $\mathbf{Q} \cdot \text{NTT}(i_b) = \text{NTT}(i)$, where $\mathbf{Q} = \begin{bmatrix} 1 & 2 & 4 & \dots & 2^{d-1} \\ 1 & 2 & 4 & \dots & 2^{d-1} \\ \vdots & \vdots & & & \vdots \\ 1 & 2 & 4 & \dots & 2^{d-1} \end{bmatrix}$. This means that all of $i$'s NTT coefficients are equal, hence $i$ is an integer.

All in all, $i$ is an integer whose binary representation has length $d$, i.e $i \in \mathcal{I}$.

# Verifiable arbitrary-message encryption almost for free

We add an extra commitment to $\sqrt{q}i$ and use the commitment to $i$, $\sqrt{q}i$ as the ciphertext.

$$\begin{bmatrix} \mathbf{f}_0 \\ f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_0 \\ \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix} \mathbf{r} + \begin{bmatrix} \mathbf{0} \\ i \\ \sqrt{q}i \end{bmatrix}.$$

# Verifiable arbitrary-message encryption almost for free

We add an extra commitment to $\sqrt{q}i$ and use the commitment to $i$, $\sqrt{q}i$ as the ciphertext.

$$\begin{bmatrix} \mathbf{f}_0 \\ f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_0 \\ \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix} \mathbf{r} + \begin{bmatrix} \mathbf{0} \\ i \\ \sqrt{q}i \end{bmatrix}.$$

The setup authority plants a decryption key as follows

$$\mathbf{a}_1^T = \mathbf{h}_1^T \mathbf{A}_0 + \mathbf{e}_1^T,$$

$\mathbf{a}_2^T = \mathbf{h}_2^T \mathbf{A}_0 + \mathbf{e}_2^T$, for small $\mathbf{h}_1, \mathbf{h}_2, \mathbf{e}_1, \mathbf{e}_2$.

The new $\mathbf{a}_1, \mathbf{a}_2$ are indistinguishable from uniform under the Module-LWE assumption, therefore this change is secure.

# Decryption

Given the ciphertext

$$\begin{bmatrix} \mathbf{f}_0 \\ f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_0 \\ \mathbf{h}_1^T \mathbf{A}_0 + \mathbf{e}_1^T \\ \mathbf{h}_2^T \mathbf{A}_0 + \mathbf{e}_2^T \end{bmatrix} \mathbf{r} + \begin{bmatrix} \mathbf{0} \\ i \\ \sqrt{q}i \end{bmatrix},$$

decryption is as follows :

# Decryption

Given the ciphertext

$$\begin{bmatrix} \mathbf{f}_0 \\ f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_0 \\ \mathbf{h}_1^T\mathbf{A}_0 + \mathbf{e}_1^T \\ \mathbf{h}_2^T\mathbf{A}_0 + \mathbf{e}_2^T \end{bmatrix} \mathbf{r} + \begin{bmatrix} \mathbf{0} \\ i \\ \sqrt{q}i \end{bmatrix},$$

decryption is as follows :

- Compute $x_1 = f_1 - \mathbf{h}_1^T\mathbf{f}_0$
- Compute $x_2 = f_2 - \mathbf{h}_2^T\mathbf{f}_0$
- Compute $k = (\sqrt{q}x_1 - x_2) \mod \sqrt{q}$ <- $k = (\sqrt{q}\mathbf{e}_1^T - \mathbf{e}_2^T)\mathbf{r}$
- return $(x_2 + k)/\sqrt{q}$ <- if $\mathbf{e}_1, \mathbf{e}_2$ small enough then $(x_2 + k)/\sqrt{q} = i$.

# Well-formedness ZK proof for almost free

To prove that the ciphertext $\mathbf{f}_0, f_1, f_2$ is well formed, the user needs to prove that

1. he knows a short $\mathbf{r}$ such that

$$\begin{bmatrix} \mathbf{f}_0 \\ f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_0 \\ \mathbf{a}_1^T \\ \mathbf{a}_2^T \end{bmatrix} \mathbf{r} + \begin{bmatrix} \mathbf{0} \\ i \\ i' \end{bmatrix}$$

# Well-formedness ZK proof for almost free

To prove that the ciphertext $\mathbf{f}_0, f_1, f_2$ is well formed, the user needs to prove that

1. he knows a short $\mathbf{r}$ such that

$$\begin{bmatrix} \mathbf{f}_0 \\ f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_0 \\ \mathbf{a}_1^T \\ \mathbf{a}_2^T \end{bmatrix} \mathbf{r} + \begin{bmatrix} \mathbf{0} \\ i \\ i' \end{bmatrix}$$

2. The messages $i, i'$ in $f_1, f_2$ respectively are such that $i' = \sqrt{q}i$.

# Well-formedness ZK proof for almost free

To prove that the ciphertext $\mathbf{f}_0, f_1, f_2$ is well formed, the user needs to prove that

1. he knows a short $\mathbf{r}$ such that

$$\begin{bmatrix} \mathbf{f}_0 \\ f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_0 \\ \mathbf{a}_1^T \\ \mathbf{a}_2^T \end{bmatrix} \mathbf{r} + \begin{bmatrix} \mathbf{0} \\ i \\ i' \end{bmatrix}$$

2. The messages $i, i'$ in $f_1, f_2$ respectively are such that $i' = \sqrt{q}i$.

1) is proven already and 2) is cheap (about 2KB extra to the signature).

# Unreliable decryption

The soundness of the well-formedness proof only ensures that there exists $\bar{\mathbf{r}}$ slightly bigger than $\mathbf{r}$ and $\bar{c} \in \mathcal{C}$ such that

$$\bar{c} \begin{bmatrix} \mathbf{f}_0 \\ f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_0 \\ \mathbf{a}_1^T \\ \mathbf{a}_2^T \end{bmatrix} \bar{\mathbf{r}} + \bar{c} \begin{bmatrix} \mathbf{0} \\ i \\ i' \end{bmatrix}.$$

# Unreliable decryption

The soundness of the well-formedness proof only ensures that there exists $\bar{\mathbf{r}}$ slightly bigger than $\mathbf{r}$ and $\bar{c} \in \mathcal{C}$ such that

$$\bar{c} \begin{bmatrix} \mathbf{f}_0 \\ f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_0 \\ \mathbf{a}_1^T \\ \mathbf{a}_2^T \end{bmatrix} \bar{\mathbf{r}} + \bar{c} \begin{bmatrix} \mathbf{0} \\ i \\ i' \end{bmatrix}.$$

The verifier is not sure that the decryption is correct !

He only knows that there exists $c'$ such that with $\bar{c} = c - c'$, $\bar{c}\mathbf{f}$ is a valid ciphertext for the message $\bar{c}i$.

# Verifiable decryption

# Verifiable decryption

## Actual decryption algorithm

- Sample $c' \leftarrow \mathcal{C}$
- $\bar{c} = c - c'$
- Compute $x_1 = f_1 - \mathbf{h}_1^T \mathbf{f}_0$ (Try and decrypt $\bar{c}(\mathbf{f}_0, f_1, f_2)$)
- Compute $x_2 = f_2 - \mathbf{h}_2^T \mathbf{f}_0$
- Compute $k = \bar{c}(\sqrt{q}x_1 - x_2) \mod \sqrt{q}$
- If $\|\bar{c}(\sqrt{q}x_1 - x_2)\|_\infty \leq \frac{q}{4\|\bar{c}\|}$, then
- return $(x_2 + k)/(\sqrt{q}\bar{c})$ (Divide the plaintext by $\bar{c}$).

# Verifiable decryption

## Actual decryption algorithm

- Sample $c' \leftarrow \mathcal{C}$
- $\bar{c} = c - c'$
- Compute $x_1 = f_1 - \mathbf{h}_1^T \mathbf{f}_0$ (Try and decrypt $\bar{c}(\mathbf{f}_0, f_1, f_2)$)
- Compute $x_2 = f_2 - \mathbf{h}_2^T \mathbf{f}_0$
- Compute $k = \bar{c}(\sqrt{q}x_1 - x_2) \mod \sqrt{q}$
- If $\|\bar{c}(\sqrt{q}x_1 - x_2)\|_\infty \leq \frac{q}{4\|\bar{c}\|}$, then
- return $(x_2 + k)/(\sqrt{q}\bar{c})$ (Divide the plaintext by $\bar{c}$).

We prove that correctness follows from the condition and the correctness of the original decryption algorithm, and that the condition is met with non-negligible probability.

# Open question

For the traceability reduction, we need to add a uniformly random copy of $\mathbf{B}$ which we call $\mathbf{B}'$. The public matrix becomes $[\mathbf{A}|\mathbf{B}|\mathbf{B}']$, which costs about 31KB in the signature.

# Open question

For the traceability reduction, we need to add a uniformly random copy of $\mathbf{B}$ which we call $\mathbf{B}'$. The public matrix becomes $[\mathbf{A}|\mathbf{B}|\mathbf{B}']$, which costs about 31KB in the signature.

Adding $\mathbf{B}'$ seems to be an artefact of the proof, and removing it does not seem to affect security. It is a very good open question to find a proof that does not rely on a copy of $\mathbf{B}$ (without affecting other parameters) to reduce the signature (and public key) size.

# Bibliography I



Thomas Attema, Vadim Lyubashevsky, and Gregor Seiler.
Practical product proofs for lattice commitments.
Cryptology ePrint Archive, Report 2020/517, 2020.
https://ia.cr/2020/517.


Jonathan Bootle, Vadim Lyubashevsky, and Gregor Seiler.
Algebraic techniques for short(er) exact lattice-based zero-knowledge proofs.
Cryptology ePrint Archive, Report 2019/642, 2019.
https://ia.cr/2019/642.


Rafael del Pino, Vadim Lyubashevsky, and Gregor Seiler.
Lattice-based group signatures and zero-knowledge proofs of automorphism stability.
Cryptology ePrint Archive, Report 2018/779, 2018.
https://ia.cr/2018/779.


Muhammed F. Esgin, Ngoc Khanh Nguyen, and Gregor Seiler.
Practical exact proofs from lattices: New techniques to exploit fully-splitting rings.
Cryptology ePrint Archive, Report 2020/518, 2020.
https://ia.cr/2020/518.


Muhammed F. Esgin, Ron Steinfeld, Joseph K. Liu, and Dongxi Liu.
Lattice-based zero-knowledge proofs: New techniques for shorter and faster constructions and applications.
Cryptology ePrint Archive, Report 2019/445, 2019.
https://ia.cr/2019/445.


Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler.
Shorter lattice-based zero-knowledge proofs via one-time commitments.
Cryptology ePrint Archive, Report 2020/1448, 2020.
https://ia.cr/2020/1448.