CryptanalYSIS of GGH Multilinear Maps

Jung Hee Cheon, Changmin Lee

Seoul National University

October 25, 2017
Toolkit for \( \mathbb{Z}[X]/\langle X^n + 1 \rangle \)
Definition of NTRU problem

Definition ((Generalized) NTRU Problem)

Let $D \ll q, \tau$ be a real number and $q \in \mathbb{Z}$. $NTRU_{R,q,D,\tau}$ is the problem of, given $h = [f/g]_q$ for $\|f\|, \|g\| < D$, finding nonzero elements $a \cdot f$ and $a \cdot g$ s.t. $\|a \cdot f\|, \|a \cdot g\| < \tau$.

Here

- $R := \mathbb{Z}[X]/\langle X^n + 1 \rangle$, $[R]_q := \mathbb{Z}_q[X]/\langle X^n + 1 \rangle$, $\mathbb{Z}_q := (-q/2, q/2) \cap \mathbb{Z}$

- $\|u\| = \sqrt{\sum_{i=0}^{n-1} u_i^2}$ for $u = \sum_{i=0}^{n-1} u_i X^i \in R$

- We set $n = 2^k \approx \lambda^2$, $q = 2^\lambda$, $D = \sqrt{n}$, $\tau = q/D$. 
**Basic idea**

**Strategy**: Find \( c \in \mathbb{R} \) s.t. \( \| [c]_q \| \) and \( \| [c \cdot h]_q \| \) are small for \( h = [g/f]_q \).

If \( g \) and \( f \) are relative prime, \((c, [c \cdot h]_q)\) is a multiple of \((f, g)\).

\[ \therefore \text{Let } d := [c \cdot h]_q. \] Then,

\[
\begin{align*}
[c \cdot g]_q &= [d \cdot f]_q \\
\Rightarrow c \cdot g &= d \cdot f \in \langle f \rangle \\
\Rightarrow c &= a \cdot f \text{ for some } a \in \mathbb{R} \text{ and } d = a \cdot g \\
\Rightarrow (c, d) &= (a \cdot f, a \cdot g)
\end{align*}
\]
Basic idea

- How to find \( c \in \mathbb{R} \) s.t. \( \| [c]_q \| \) and \( \| [c \cdot h]_q \| \) are small.

- The answer is to find a short vector of the following (row) lattice:

\[
\Lambda^q_h = \begin{bmatrix}
I_n & H \\
0 & q \cdot I_n
\end{bmatrix}.
\]

Where \( H = (h_{ij}) \) and \( h_{ij} \) is a \( j \)-th coefficient of \( h \cdot X^i \).
Basic idea

- **Vec**: $R \to \mathbb{Z}^n$ is defined by $\text{Vec}(u) = (u_0, \cdots, u_{n-1})$ for $u = \sum_{i=0}^{n-1} u_i X^i$

- Consider

  $$\Lambda_h = \begin{bmatrix} I_n & H \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{n-1} \\ 1 & \cdots & 0 & -h_{n-1} & h_0 & \cdots & h_{n-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & -h_1 & -h_2 & \cdots & h_0 \end{bmatrix}.$$

- Then, we have

  $$\text{vec}(c) \cdot \begin{bmatrix} I_n & H \end{bmatrix} = (\text{vec}(c), \text{vec}(c \cdot h)) \iff (c, c \cdot h)$$
Basic idea

- To obtain \([\text{vec}(c)]_q [\text{vec}(c \cdot h)]_q\) instead of \([\text{vec}(c) \text{ vec}(c \cdot h)]\), we define the row lattice:

\[
\Lambda_h^q = \begin{bmatrix}
I_n & H \\
q \cdot I_n & 0 \\
0 & q \cdot I_n
\end{bmatrix} = \begin{bmatrix}
I_n & H \\
0 & q \cdot I_n
\end{bmatrix}.
\]

- Hence if one can find a short vector of \(\Lambda_h^q\), the NTRU problem is solved.

However, when the dimension is large, \(c\) must be large and it takes a long time to find a short vector. Hence, reducing the dimension leads us to solve the NTRU problem better.
Consider a Norm map, \( N : K_0 \to K_1 \), \( N(a) = a \cdot a(-X) \)
where \( K_i = \mathbb{Q}[X^{2^i}] / \langle X^n + 1 \rangle \), \( K_0/K_1 \) is a Galois extension.

The NTRU problem with \( h = \begin{bmatrix} g \\ f \end{bmatrix} \) is to find \( (a \cdot f, a \cdot g) \).

\[ \Rightarrow \text{For } N(h) = \begin{bmatrix} N(g) \\ N(f) \end{bmatrix} \] is to find \( (a' \cdot N(f), a' \cdot N(g)) \).

If one can recover \( a' \cdot N(f) := d \cdot f \) with small \( d \in R \), we have
\[ [d \cdot f \cdot h]_q = [d \cdot g]_q = d \cdot g. \]

\( \Rightarrow (d \cdot f, d \cdot g) \) is a solution of NTRU with \( h \).
Definition of PIP problem

Definition (PIP Problem)
Given $h_i = f \cdot c_i$ for fixed $f \in R$, Principal ideal problem, $PIP_R$ is the problem is to find an element $u \cdot f$ for an unit $u \in R$.

Definition (SPIP Problem)
Let $\tau'$ be a real number. Given $h_i = f \cdot c_i$ for fixed $f \in R$, $SPIP_{R,\tau}$ is the problem to find an element $u \cdot f$ for an unit $u \in R$. s.t. $\|u \cdot f\| < \tau'$.

Here

- $R := \mathbb{Z}[X]/\langle X^n + 1 \rangle$, $\|u\| = \sqrt{\sum_{i=0}^{n-1} u_i^2}$ for $u = \sum_{i=0}^{n-1} u_i X^i \in R$
- We set $n = 2^k \approx \lambda^2$ $\tau' = n^2$. 

Changmin Lee  
Analysis of Multilinear Maps  
October 25, 2017 9 / 30
Current Results

- NTRU is solved in $2^{\Theta(n \log M / \log^2 q)} = 2^{\Theta(\lambda^2 \log \lambda / \lambda^2)} = \text{poly}(\lambda)$ time\(^1\) (When $n$ becomes large, this algorithm takes a long time)

- PIP is solved
  - in $2^{n^{2/3}}$ time with classical computing\(^2\)
  - in $\text{poly}(n)$ time with quantum computing.\(^3\)

- SPIP is solved with PIP algorithm and shortening.
  - The shortening is done in $\text{poly}(n)$ and outputs $2^\tilde{O}(\sqrt{n})$.
  - If $f$ is sampled from Gaussian distribution and small. One can find $f$\(^4\).

---

\(^1\)[CHL17], Cryptanalysis of Middle Lattice on the Overstretched NTRU Problem for General Modulus Polynomial

\(^2\)[BF14], Subexponential class group and unit group computation in large degree number fields.

\(^3\)[BS16], A polynomial time quantum algorithm for computing class groups and solving the principal ideal problem in arbitrary degree number fields.

\(^4\)[CDPR16], Recovering Short Generators of Principal Ideals in Cyclotomic Rings
Analysis of
GGH Multilinear map
We consider the ring:

\[ R = \mathbb{Z}[X]/\langle X^n + 1 \rangle \]

with a power of two \( n \).

Setup:
- Ciphertext space is the ring \( C = R_q = R/\langle q \rangle \) for a large prime \( q \).
- Plaintext space is the field \( P := R/\langle g \rangle \).
- Here, \( g = \sum_{i=0}^{n-1} g_i \cdot X^i \in R \) satisfies \( \|g\| = \sqrt{\sum g_i^2} \ll q \) & \( N(\langle g \rangle) \) is prime.
A level-$t$ encoding of $m \in \mathcal{P}$ is of the form:

$$\text{enc}_t(m) = \frac{m'}{z^t} \mod q,$$

with $m' = m \mod \langle g \rangle$ and $\|m'\| \leq O(n^{2t})$.

A zerotesting parameter made public is of the form:

$$p_{zt} = h \cdot z^\kappa \mod q$$

Given a level-$\kappa$ encoding $c = \frac{r'g + m}{z^\kappa}$, and $p_{zt} = \frac{h \cdot z^\kappa}{g}$,

$$[p_{zt} \cdot c]_q = \begin{cases} 
\|h \cdot r'\| & \leq q^{3/4} \quad \text{if } m = 0 \\
\|h \cdot (mg^{-1} + r')\|_q & \text{otherwise} \quad \text{if } m \neq 0 
\end{cases}$$
Hardness problems

For simplicity, we only discuss the case of $\kappa = 2$.

For a given set of public parameters,

$$\{R, q, y, x_i, p_{zt}\} \text{ and } \{\text{enc}_1(m_0), \text{enc}_1(m_1), \text{enc}_1(m_2)\},$$

the hardness problem of the GGH multilinear map is to distinguish

$$\text{enc}_2(m_0 \cdot m_1 \cdot m_2) \text{ or } \text{enc}_2(r \cdot m_1 \cdot m_2)$$

Remark: We do not know the values $g, h, z, m_i$.
To compute these secret values are also one way to solve the hardness problems of GGH multilinear map.
Hardness problems

For simplicity, we only discuss the case of $\kappa = 2$.

For a given set of public parameters,

$$\{R, q, y, x_i, p_{zt}\} \quad \text{and} \quad \{\text{enc}_1(m_0), \text{enc}_1(m_1), \text{enc}_1(m_2)\},$$

the hardness problem of the GGH multilinear map is to distinguish

$$\text{enc}_2(m_0 \cdot m_1 \cdot m_2) \quad \text{or} \quad \text{enc}_2(r \cdot m_1 \cdot m_2)$$

Remark: We do not know the values $g, h, z, m_i$. To compute these secret values are also one way to solve the hardness problems of GGH multilinear map.
We will exploit the algorithm mentioned above with the following properties.

- The size of numerator of level-1 encodings is small.

\[
\frac{\text{enc}_1(m_1)}{\text{enc}_1(m_2)} = \frac{m'_1}{z} \frac{z}{m'_2} = \frac{m'_1}{m'_2} \mod q, \quad \text{with } \|m'_i\| \leq n^2
\]

If one can recover \(m'_i\), it reveals \(z\).

- The zerotesting value of a top level encoding of zero is in \(R\), not \(R_q\).

\[
[p_{zt} \cdot c]_q = h \cdot r' \quad \text{for } c = [r' \cdot g/z^\kappa]_q
\]

Collecting the several quantities, \(h\) can be recovered.
1. We can compute many small ratio:

\[ \frac{y}{x_i} = \left[ \frac{(1 + rg)}{r_i g} \right]_q. \]

2. Using an algorithm for NTRU, we can recover a relatively small multiple pair, \((c_i \cdot (1 + rg), c_i \cdot r_i g)\).

3. From the several values \(c_i \cdot (1 + rg)\), one can recover \(1 + rg\) with quantum PIP algorithm. Then, we can also recover \(z\) and set \(\text{enc}_2(m_0 \cdot m_1 \cdot m_2) = z \cdot \text{enc}_1(m_0) \cdot \text{enc}_1(m_1) \cdot \text{enc}_1(m_2)\).

4. Compute \(\left[ p_{zt} \cdot (\text{enc}_2(m_0 \cdot m_1 \cdot m_2) - \text{enc}_2(r \cdot m_1 \cdot m_2)) \right]_q\). If \(r = m_0\) it is small. If not, it is large.
1. We can compute many small ratio:

\[ \frac{y}{x_i} = \left[ \frac{(1 + rg)}{r_i g} \right]_q. \]

2. Using an algorithm for NTRU, we can recover a relatively small multiple pair, \((c_i \cdot (1 + rg), c_i \cdot r_i g)\).

3. From the several values \(c_i \cdot (1 + rg)\), one can recover \(1 + rg\) with quantum PIP algorithm. Then, we can also recover \(z\) and set

\[ \text{enc}_2(m_0 \cdot m_1 \cdot m_2) = z \cdot \text{enc}_1(m_0) \cdot \text{enc}_1(m_1) \cdot \text{enc}_1(m_2). \]

4. Compute \([p_{zt} \cdot (\text{enc}_2(m_0 \cdot m_1 \cdot m_2) - \text{enc}_2(r \cdot m_1 \cdot m_2))]_q\). If \(r = m_0\) it is small. If not, it is large.
First attack

1. We can compute many small ratio:

\[ \frac{y}{x_i} = \left( \frac{(1 + rg)}{r_i g} \right)_q. \]

2. Using an algorithm for NTRU, we can recover a relatively small multiple pair, \((c_i \cdot (1 + rg), c_i \cdot r_i g)\).

3. From the several values \(c_i \cdot (1 + rg)\), one can recover \(1 + rg\) with quantum PIP algorithm. Then, we can also recover \(z\) and set \(\text{enc}_2(m_0 \cdot m_1 \cdot m_2) = z \cdot \text{enc}_1(m_0) \cdot \text{enc}_1(m_1) \cdot \text{enc}_1(m_2)\).

4. Compute \[p_{zt} \cdot (\text{enc}_2(m_0 \cdot m_1 \cdot m_2) - \text{enc}_2(r \cdot m_1 \cdot m_2))\] \(q\). If \(r = m_0\) it is small. If not, it is large.
Second attack

1. Compute many valid zerotesting values:

\[
\begin{align*}
\mathbf{v}_{ij} &= p_{zt} \cdot x_i \cdot \text{enc}_1(m_j) = r_i \cdot m'_j \cdot h \in R \\
\mathbf{u}_{ij} &= p_{zt} \cdot x_i \cdot x_j = r_i \cdot r_j \cdot h \cdot g \in R
\end{align*}
\]

These are equation over \( R!! \)

2. We can recover a generator of ideal \( \langle h \rangle \) and \( \langle g \cdot h \rangle \) by collecting values with quantum PIP computing.

3. From the generator of \( \langle h \rangle \) and \( \langle g \cdot h \rangle \), we can also recover a generator of \( \langle g \rangle \).
Second attack

1. Compute many valid zero-testing values:

\[ v_{ij} = p_{zt} \cdot x_i \cdot \text{enc}_1(m_j) = r_i \cdot m'_j \cdot h \in R \]

\[ u_{ij} = p_{zt} \cdot x_i \cdot x_j = r_i \cdot r_j \cdot h \cdot g \in R \]

These are equation over \( R!! \)

2. We can recover a generator of ideal \( \langle h \rangle \) and \( \langle g \cdot h \rangle \) by collecting values with quantum PIP computing.

3. From the generator of \( \langle h \rangle \) and \( \langle g \cdot h \rangle \), we can also recover a generator of \( \langle g \rangle \).
Second attack

The next step is to recover the message $m_j$.

Hereafter we write $x$ instead of $x_i$ since we only need one encoding of zero.

1. Remove modular $q$ by zerotesting:

$$M_j = p_{zt} \cdot \text{enc}_1(m_j) \cdot x = m_j \cdot r_i \cdot h \in R$$

$$A = p_{zt} \cdot y \cdot x = (1 + r \cdot g) \cdot r_i \cdot h \in R$$

2. Compute $m_j^* = M_j \cdot A^{-1} \mod \langle g \rangle \in R$, which satisfies

$$m_j^* = m_j \mod \langle g \rangle \text{ and small.}$$

3. Compute $[p_{zt} \cdot ((m_0^* \cdot m_1^* \cdot m_2^*) \cdot y^2 - \text{enc}_2(r \cdot m_1 \cdot m_2))]_q$.

If $r = m_0$ it is small. If not, it is large.
The next step is to recover the message $m_j$.
Hereafter we write $x$ instead of $x_i$ since we only need one encoding of zero.

1. Remove modular $q$ by zerotesting:

$$M_j = p_{zt} \cdot \text{enc}_1(m_j) \cdot x = m_j \cdot r_i \cdot h \in R$$
$$A = p_{zt} \cdot y \cdot x = (1 + r \cdot g) \cdot r_i \cdot h \in R$$

2. Compute $m_j^* = M_j \cdot A^{-1} \mod \langle g \rangle \in R$, which satisfies

$$m_j^* = m_j \mod \langle g \rangle \text{ and } small.$$  

3. Compute $[p_{zt} \cdot ((m_0^* \cdot m_1^* \cdot m_2^*) \cdot y^2 - \text{enc}_2(r \cdot m_1 \cdot m_2))]_q$.
If $r = m_0$ it is small. If not, it is large.
In summary

- Applying the first algorithm to GGH13 with \( y, x_i \), it is broken in polynomial time.
  
  This attack can be prevented by setting a large parameter.

- Applying the second algorithm to GGH13 with \( y, x_i \), it is broken in (classical) polynomial time\(^5\).

- The security of GGH13 without \( y, x_i \) is still a hard problem with (classical) computing.

\(^5\)[HJ16], Cryptanalysis of GGH Map
Modified GGH Multilinear maps
New Approach

Intuition:

- Current attacks can recover $h$ and $g$ from the zerotesting value $h \cdot r'$.

- If there is a small error in the zerotesting values, the attacks cannot be applied but zerotesting still works.

It means that the zerotesting value of $\text{enc}_\kappa(m)$ is modified

from $[h \cdot m \cdot g^{-1} + h \cdot r']_q$ to $[h \cdot m \cdot g^{-1} + h \cdot r' + e]_q$,

where $e$ is a small element of $R$.

- To add small error values, we consider the bit decomposition $B.D$ and the power with error $Pow$ functions.
New Approach

Intuition:

- Current attacks can recover $h$ and $g$ from the zerotesting value $h \cdot r'$.
- If there is a small error in the zerotesting values, the attacks cannot be applied but zerotesting still works.

It means that the zerotesting value of $\text{enc}_\kappa(m)$ is modified

$$\text{from } [h \cdot m \cdot g^{-1} + h \cdot r']_q \text{ to } [h \cdot m \cdot g^{-1} + h \cdot r' + e]_q,$$

where $e$ is a small element of $R$.

- To add small error values, we consider the bit decomposition $B.D$ and the power with error $\text{Pow}$ functions.
New Approach

Intuition:
- Current attacks can recover \( h \) and \( g \) from the zero-testing value \( h \cdot r' \).
- If there is a small error in the zero-testing values, the attacks cannot be applied but zero-testing still works.

It means that the zero-testing value of \( \text{enc}_\kappa(m) \) is modified

\[
\text{from } [h \cdot m \cdot g^{-1} + h \cdot r']_q \text{ to } [h \cdot m \cdot g^{-1} + h \cdot r' + e]_q,
\]

where \( e \) is a small element of \( R \).
- To add small error values, we consider the bit decomposition \( B.D \) and the power with error \( \text{Pow} \) functions.
New Approach

Notation:

- For a fixed element \( s \in R_q \), we define a Bit decomposition map \( B.D_s : R_q \rightarrow R_q^\ell \) and a power map \( Pow_s : R_q \rightarrow R_q^{3\ell} \) as

\[
B.D_s(a) = (a_0, a_1, \cdots, a_{\ell-1}) \quad \text{for} \quad a = \sum a_i \cdot s^i
\]

\[
Pow_s(b) = (b + e_0, s \cdot b + e_1, \cdots, s^{3\ell-1} \cdot b + e_{3\ell-1}),
\]

where \( e_i \) is a small element of \( R \).

Remark:
The expression of \( B.D \) is not unique. For example

\[
(a_0 + a_1 \cdot s, 0, a_2, \cdots, a_{\ell-1}) \quad \text{and} \quad (a, 0, \cdots, 0)
\]

are also \( B.D_s(a) \).
Notation:

For a fixed element $s \in R_q$, we define a Bit decomposition map $B.D_s : R_q \rightarrow R_q^\ell$ and a power map $Pow_s : R_q \rightarrow R_q^{3\ell}$ as

$$B.D_s(a) = (a_0, a_1, \ldots, a_{\ell-1}) \text{ for } a = \sum a_i \cdot s^i$$

$$Pow_s(b) = (b + e_0, s \cdot b + e_1, \ldots, s^{3\ell-1} \cdot b + e_{3\ell-1})$$

where $e_i$ is a small element of $R$.

Remark:
The expression of $B.D$ is not unique. For example

$$(a_0 + a_1 \cdot s, 0, a_2, \ldots, a_{\ell-1}) \text{ and } (a, 0, \ldots, 0)$$

are also $B.D_s(a)$. 
From now on, we identify an arbitrary vector $\vec{v} \in \mathbb{R}^i$, for $i \leq 3\ell$ to a vector $(\vec{v} \parallel \vec{0}) \in \mathbb{R}^{3\ell}$.

With the above notation abusing, we have:

$$\langle B.D_s(a), \ Pow_s(b) \rangle = a \cdot b + e^*,$$

where the size of $e^*$ depends on that of $a_i$ and $e_i$. 

New Encoding

Setup:
- An algebraic setup is same to that of GGH13.
- Let $s$ be a secret small element of $R$.
- A level-$t$ encoding of $m \in \mathcal{P}$ is of the form:

$$\text{enc}_t(m) = \left[ \frac{B \cdot D_s(m')}{{z^t}} \right]_q$$

with $m' = m \mod \langle g \rangle$,

where $m'$ and $\| B \cdot D_s(m') \|$ are small.
Addition & Multiplication

Setup:

- Suppose $c \in R$, $\text{enc}_1(m_1)$, and $\text{enc}_1(m_2)$ are given.

We need the following homomorphic properties to construct a multilinear map

\[
\begin{align*}
    c \cdot \text{enc}_1(m_1) &= \text{enc}_1(c \cdot m_1) \\
    \text{enc}_1(m_1) + \text{enc}_1(m_2) &= \text{enc}_1(m_1 + m_2) \\
    \text{enc}_1(m_1) \cdot \text{enc}_1(m_2) &= \text{enc}_2(m_1 \cdot m_2)
\end{align*}
\]

Definition of arithmetics among $B.D$ values are required.
Setup:

- Suppose $c \in R$, $\text{enc}_1(m_1)$, and $\text{enc}_1(m_2)$ are given.

We need the following homomorphic properties to construct a multilinear map

\[
\begin{align*}
    c \cdot \text{enc}_1(m_1) &= \text{enc}_1(c \cdot m_1) \\
    \text{enc}_1(m_1) + \text{enc}_1(m_2) &= \text{enc}_1(m_1 + m_2) \\
    \text{enc}_1(m_1) \cdot \text{enc}_1(m_2) &= \text{enc}_2(m_1 \cdot m_2)
\end{align*}
\]

Definition of arithmetics among $B.D$ values are required.
Setup:

- Suppose $B.D_s(a) = (a_0, \cdots, a_{\ell-1})$ and $B.D_s(b) = (b_0, \cdots, b_{\ell-1})$ are given. i.e. $a = \sum a_i \cdot s^i$ and $b = \sum b_i \cdot s^i$.

With simple calculations,

\[
\begin{align*}
c \cdot a &= \sum_i (c \cdot a_i) \cdot s^i \\
a + b &= \sum_i (a_i + b_i) \cdot s^i \\
a \cdot b &= \sum_i \sum_{j+k=i} (a_j \cdot b_k) \cdot s^i
\end{align*}
\]
So we define the following arithmetics:

\[ c \cdot B.D_s(a) := (c \cdot a_0, c \cdot a_1, \cdots, c \cdot a_{\ell-1}) \]
\[ = B.D_s(c \cdot a) \]
\[ B.D_s(a) + B.D_s(b) := (a_0 + b_0, a_1 + b_1, \cdots, a_{\ell-1} + b_{\ell-1}) \]
\[ = B.D_s(a + b) \]
\[ B.D_s(a) \ast B.D_s(b) := (a_0 \cdot b_0, a_1 \cdot b_0 + a_0 \cdot b_1, \cdots, \sum_{j+k=2\ell-2} a_j b_k) \]
\[ = B.D_s(a \cdot b) \]

The arithmetics of \( \text{enc}_1(m) = \left[ \frac{B.D_s(m')}{z} \right]_q \) can be naturally extended by the arithmetics of \( B.D \).
Addition & Multiplication

So we define the following arithmetics:

\[ c \cdot B.D_s(a) := (c \cdot a_0, c \cdot a_1, \cdots, c \cdot a_{\ell-1}) \]
\[ = B.D_s(c \cdot a) \]

\[ B.D_s(a) + B.D_s(b) := (a_0 + b_0, a_1 + b_1, \cdots, a_{\ell-1} + b_{\ell-1}) \]
\[ = B.D_s(a + b) \]

\[ B.D_s(a) \ast B.D_s(b) := (a_0 \cdot b_0, a_1 \cdot b_0 + a_0 \cdot b_1, \cdots, \sum_{j+k=2\ell-2} a_j b_k) \]
\[ = B.D_s(a \cdot b) \]

The arithmetics of \( \text{enc}_1(m) = \left[ \frac{B.D_s(m')}{z} \right]_q \) can be naturally extended by the arithmetics of \( B.D \).
Zerotesting

Setup:
- A zerotesting parameter made public is of the form:

\[ p_{zt} = z^3 \cdot \text{Pow}_s \left( \frac{h}{g} \mod q \right) \]

Given a level-\(\kappa = 3\) encoding \(c = \frac{B.D_s(r'g + m)}{z^3}\), and \(p_{zt}\),

\[ \langle c, p_{zt} \rangle = \begin{cases} 
\| h \cdot r + e \| & \leq q^{3/4} \quad \text{if } m = 0 \\
\|[h \cdot (mg^{-1} + r') + e]_q\| & \text{otherwise} \quad \text{if } m \neq 0 
\end{cases} \]
Hardness problem

We know the following:

- public parameters of GGH, such as $n, q$
- level-1 encodings of 1 and 0: $y$ and $x_i$
- zerotesting parameter $p_{zt} = z^\kappa \cdot Pow_s(h/g)$
- encodings of messages: $enc_1(m_i)$ for $0 \leq i \leq \kappa$

We want to know:

- $\left[ p_{zt} \cdot enc_{\kappa} \left( \prod_{i=0}^{\kappa} m_i \right) \right]_q$
- Secret parameters $g, h, z, s$
Thank you for your attention.