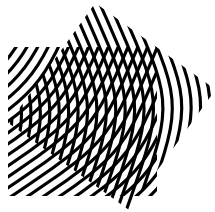
A Framework for Cryptographic Problems from Linear Algebra

Carl Bootland, Wouter Castryck,
Alan Szepieniec and Frederik Vercauteren

Dept. of Electrical Engineering, COSIC
KU Leuven



COSIC

Post-Quantum Cryptography Standardization Process


NIST

Search CSRC

CSRC MENU

Information Technology Laboratory

COMPUTER SECURITY RESOURCE CENTER

CSRC

PROJECTS | POST-QUANTUM CRYPTOGRAPHY

Post-Quantum Cryptography

Post-Quantum Cryptography Standardization

The Round 2 candidates were announced January 30, 2019. NISTIR 8240, Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process is now available.

Call for Proposals Announcement *(information retained for historical purposes-call closed 11/30/2017)*

NIST has initiated a process to solicit, evaluate, and standardize one or more quantum-resistant public-key cryptographic algorithms. Currently, public-key cryptographic algorithms are specified in FIPS 186-4, *Digital Signature Standard*, as well as special publications SP 800-56A Revision 2, *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography* and SP 800-56B Revision 1, *Recommendation for Pair-Wise Key-Establishment Schemes Using Integer Factorization Cryptography*. However, these algorithms are vulnerable to attacks from large-scale quantum computers (see NISTIR 8105, *Report on Post Quantum Cryptography*).

It is intended that the new public-key cryptography standards will specify one or more additional unclassified, publicly disclosed digital signature, public-key encryption, and key-establishment algorithms that are available worldwide, and are capable of protecting sensitive government information well into the foreseeable future, including after the advent of quantum

PROJECT LINKS

Overview
FAQs
News
Events
Publications
Presentations

ADDITIONAL PAGES

Post-Quantum Cryptography Standardization
Call for Proposals
Example Files
Round 1 Submissions
Round 2 Submissions

Aim: "to solicit, evaluate, and standardize one or more quantum-resistant public-key cryptographic algorithms."

Post-Quantum Cryptography Standardization Process

Round 1 submission categorization

	Signature	KEM/Encryption	Total
Lattice Based	5	21	26
Code Based	2	17	19
Multi-variate	7	2	9
Hash based	3	0	3
Other	2	5	7

Round 2 candidates (announced January 30, 2019)

	Signature	KEM/Encryption	Total
Lattice Based	3	8	11
Code Based	0	7	7
Multi-variate	4	0	4
Hash based	1	0	1
Other	1	2	3

Learning with errors (LWE)

Problem: Solve a system of random 'noisy' linear equations

Learning with errors (LWE)

Problem: Solve a system of random 'noisy' linear equations

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_m \end{pmatrix} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & & \vdots \\ a_{i,1} & a_{i,2} & \cdots & a_{i,n} \\ \vdots & \vdots & & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix}$$

Learning with errors (LWE)

Problem: Solve a system of random 'noisy' linear equations

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_m \end{pmatrix} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & & \vdots \\ a_{i,1} & a_{i,2} & \cdots & a_{i,n} \\ \vdots & \vdots & & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_i \\ \vdots \\ e_m \end{pmatrix} \mod q$$

- e_i *small* 'errors'
- uniformly random $a_{i,j}$

Leads to schemes with large key sizes

Ring-LWE (informally)

Problem: Solve a system of structured 'noisy' linear equations

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_m \end{pmatrix} = \begin{pmatrix} & & \\ \vdots & A_1 & \vdots \\ & \vdots & \\ \vdots & A_{m/n} & \vdots \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_i \\ \vdots \\ e_m \end{pmatrix} \mod q$$

- A_i independent structured $n \times n$ matrices
  - e.g. anti-circulant

Module-LWE (informally)

Problem: Solve a system of structured 'noisy' linear equations

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_m \end{pmatrix} = \begin{pmatrix} A_{1,1} & \cdots & A_{1,r} \\ \vdots & \vdots & \vdots \\ A_{mr/n,1} & \cdots & A_{mr/n,r} \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_i \\ \vdots \\ e_m \end{pmatrix} \mod q$$

- $A_{i,j}$ independent structured $n/r \times n/r$ matrices
- r is rank of module

The Ring in Ring-LWE

- Identify the vector space $\mathbb{Z}_q^n$ with the ring

$$\mathbb{Z}_q^n \leftrightarrow R_q := \frac{\mathbb{Z}_q[x]}{(f(x))}$$

  - $f(x)$ monic of degree n

$$(s_1, s_2, \ldots, s_n)^T \leftrightarrow \mathbf{s}(x) = s_1 + s_2 x + \cdots + s_n x^{n-1}$$

- A_i are matrices of multiplication by $\mathbf{a}_i(x) \in R_q$
  - Anti-circulant matrices $\implies$ $f(x) = x^n + 1$
- We don't need q to be prime

Ring-LWE (More formally)

Ring-LWE Search problem:

- $f(x)$ irreducible
- Samples $(\mathbf{a}_i(x), \mathbf{b}_i(x)) \in R_q \times R_q$

$$\mathbf{b}_i(x) = \mathbf{a}_i(x)\mathbf{s}(x) + \mathbf{e}_i(x)$$

  - uniformly random $\mathbf{a}_i(x)$
  - uniformly random $\mathbf{s}(x)$
  - $\mathbf{e}_i(x) \leftarrow \chi$ (distribution of small elements)
- recover $\mathbf{s}(x)$

Ring-LWE (More formally)

Ring-LWE Search problem:

- $f(x)$ irreducible
- Samples $(\mathbf{a}_i(x), \mathbf{b}_i(x)) \in R_q \times R_q$

$$\mathbf{b}_i(x) = \mathbf{a}_i(x)\mathbf{s}(x) + \mathbf{e}_i(x)$$

  - uniformly random $\mathbf{a}_i(x)$
  - uniformly random $\mathbf{s}(x)$
  - $\mathbf{e}_i(x) \leftarrow \chi$ (distribution of small elements)
- recover $\mathbf{s}(x)$

- Called Poly-LWE when $s(x) \leftarrow R_q$

Interesting Submissions to the NIST Competition

Three submissions use problems which look very much like LWE but use large integer arithmetic:

- Mersenne-756839
- Ramstake
- Three Bears

Mersenne-756839 and Ramstake:

- Mersenne Low Hamming Combination (MLHC)

Three Bears:

- module version of Integer-RLWE

The Mersenne Low Hamming Combination Search Problem

- $p = 2^n - 1$ a Mersenne prime

$$\mathbb{Z}_p \leftrightarrow \{\text{bit strings of length } n\} \setminus \{11\ldots1\}$$

Problem:

- Samples $(\mathbf{a}_i, \mathbf{b}_i) \in \mathbb{Z}_p \times \mathbb{Z}_p$

$$\mathbf{b}_i = \mathbf{a}_i\mathbf{s} + \mathbf{e}_i$$

  - $\mathbf{a}_i$ uniformly random
  - $\mathbf{s}$, $\mathbf{e}_i$ Hamming weight $h \ll n$
- determine $\mathbf{s}$

The Mersenne Low Hamming Combination Search Problem

- $p = 2^n - 1$ a Mersenne prime

$$\mathbb{Z}_p \leftrightarrow \{\text{bit strings of length } n\} \setminus \{11\ldots1\}$$

Problem:

- Samples $(\mathbf{a}_i, \mathbf{b}_i) \in \mathbb{Z}_p \times \mathbb{Z}_p$

$$\mathbf{b}_i = \mathbf{a}_i\mathbf{s} + \mathbf{e}_i$$

  - $\mathbf{a}_i$ uniformly random
  - $\mathbf{s}$, $\mathbf{e}_i$ Hamming weight $h \ll n$
- determine $\mathbf{s}$

Integer-RLWE:

$$p = 2^n - 1 \rightarrow p = q^n + 1$$

Unifying the MLHC and Poly-LWE problems

Similar problems, different rings

Small elements

- Poly-LWE
  - $\mathbf{e}(x) = e_1 + e_2 x + \cdots + e_n x^{n-1}$
  - $(e_1, \ldots, e_n)$ a short vector (e.g. from spherical Gaussian)
- MLHC
  - $\mathbf{e} = e_1 + e_2 2 + \cdots + e_n 2^{n-1}$
  - $(e_1, \ldots, e_n)$ a short vector (Hamming weight h)
- Important point: coefficient vector is short
- Difference in expansion:
  - Poly-LWE: Use *explicit* base x
  - MLHC: Use *implicit* base 2

Unifying the MLHC and Poly-LWE problems

Since

$$p = 2^n - 1$$

rewrite $\mathbb{Z}_p$ as

$$\frac{\mathbb{Z}[x]}{(x^n - 1, x - 2)}$$

and R_q as

$$\frac{\mathbb{Z}[x]}{(f(x), q)}$$

Unifying the MLHC and Poly-LWE problems

Since

$$p = 2^n - 1$$

rewrite $\mathbb{Z}_p$ as

$$\frac{\mathbb{Z}[x]}{(x^n - 1, x - 2)}$$

and R_q as

$$\frac{\mathbb{Z}[x]}{(f(x), q)}$$

View $\mathbb{Z}_p$ as R_q

- $f(x) = x^n - 1$
- q replaced by $x - 2$

Three Bears

Use a Solinas prime

$$p = 2^{3120} - 2^{1560} - 1$$

hence

$$\mathbb{Z}_p \cong \frac{\mathbb{Z}[x]}{(x^{312} - x^{156} - 1, x - 2^{10})}$$

View $\mathbb{Z}_p$ as R_q

- $f(x) = x^{312} - x^{156} - 1$
- q replaced by $x - 2^{10}$

Three Bears

Use a Solinas prime

$$p = f(b)$$

hence

$$\mathbb{Z}_p \cong \frac{\mathbb{Z}[x]}{(f(x), x - b)}$$

View $\mathbb{Z}_p$ as R_q

- $f(x)$ low-degree, small coefficients
- q replaced by $x - b$

Generalising the ring

The second modulus

- Standard LWE-type problems: integer q
- Large integer arithmetic schemes: linear $x - b$
- General problem: arbitrary $g(x)$

$$R_g := \frac{\mathbb{Z}[x]}{(f(x), g(x))}$$

- $g(x)$ coprime to $f(x) \implies R_g$ finite
- Small elements defined in $R = \mathbb{Z}[x]/(f(x))$

A condition of convenience

We want the ring R_g to be easy to work with:

- Restrict possible g so that

$$(f(x), g(x)) = (a, r(x))$$

  - a an integer
  - $r(x)$ monic

- Unique representative in

$$\left\{ \alpha_0 + \alpha_1 x + \cdots + \alpha_{\deg(r)-1} x^{\deg(r)-1} \;\middle|\; \alpha_i \in \{0, 1, \ldots, a-1\} \right\}$$

- Not too restrictive
  - $6/\pi^2 \approx 60.8\%$ of randomly chosen pairs f, g
  - r linear with overwhelming probability

Other problems (Informally)

Generalise from R_q to R_g in other problems

NTRU:

- Given $\mathbf{h} \in R_q$

$$\mathbf{h} = \mathbf{u}/\mathbf{v} \mod q$$

  - $\mathbf{u}$, $\mathbf{v}$ small
- find small $\mathbf{u}'$, $\mathbf{v}'$ such that

$$\mathbf{h} = \mathbf{u}'/\mathbf{v}' \mod q$$

Other problems (Informally)

Generalise from R_q to R_g in other problems

Ring-SIS:

- Given $\mathbf{a}_1, \ldots, \mathbf{a}_m \in R_q$
  - $\mathbf{a}_i$ uniformly random
  - $m \gtrapprox n \log q$
- find small $\mathbf{z}_1, \ldots, \mathbf{z}_m$, not all zero, such that

$$\sum_{i=1}^{m} \mathbf{a}_i \mathbf{z}_i = 0$$

Why is this interesting? Security?

Lattice Attacks on LWE based primitives:

- strong lattice basis reduction
  - e.g. BKZ2.0
  - most practical attacks
  - works on integer lattices
  - find short(est) vectors
  - recover secret information
- Work in R_g but smallness defined only in $R = \mathbb{Z}[x]/(f(x))$
  - dimension depends on $\deg(f)$
  - include generators $x^i g(x) \mod f(x)$

Example Lattice: The Primal Attack on Poly-LWE

$$
\left(\begin{array}{c|c|c|c}
\multicolumn{2}{c|}{— \ \mathbf{b}_1 \ —} & \cdots & — \ \mathbf{b}_\ell \ — & w \\
\hline
\multicolumn{2}{c|}{\begin{array}{c} — \ \mathbf{a}_1 \ — \\ \vdots \\ — \ x^{n-1}\mathbf{a}_1 \bmod f \ — \end{array}} & \cdots & \begin{array}{c} — \ \mathbf{a}_\ell \ — \\ \vdots \\ — \ x^{n-1}\mathbf{a}_\ell \bmod f \ — \end{array} & \\
\hline
\multicolumn{2}{c|}{ql_n} & & & \\
\hline
\multicolumn{2}{c|}{} & \ddots & & \\
\hline
\multicolumn{2}{c|}{} & & ql_n & \\
\end{array}\right)
$$

Example Lattice: The Primal Attack on Poly-LWE

$$
\left(\begin{array}{c|c|c|c}
\multicolumn{2}{c|}{— \ \mathbf{b}_1 \ —} & \cdots & — \ \mathbf{b}_\ell \ — & w \\
\hline
\multicolumn{2}{c|}{\begin{array}{c} — \ \mathbf{a}_1 \ — \\ \vdots \\ — \ x^{n-1}\mathbf{a}_1 \bmod f \ — \end{array}} & \cdots & \begin{array}{c} — \ \mathbf{a}_\ell \ — \\ \vdots \\ — \ x^{n-1}\mathbf{a}_\ell \bmod f \ — \end{array} & \\
\hline
\multicolumn{2}{c|}{ql_n} & & & \\
\hline
\multicolumn{2}{c|}{} & \ddots & & \\
\hline
\multicolumn{2}{c|}{} & & ql_n & \\
\hline
— \ \mathbf{e}_1 \ — & & & — \ \mathbf{e}_\ell \ — & w
\end{array}\right)
$$

Example Lattice: What about for the ring R_g?

$$\left(\begin{array}{c|c|c|c}
— \ \mathbf{b}_1 \ — & \cdots & — \ \mathbf{b}_\ell \ — & w \\
\hline
— \ \mathbf{a}_1 \ — & \cdots & — \ \mathbf{a}_\ell \ — & \\
\vdots & & \vdots & \\
— \ x^{n-1}\mathbf{a}_1 \bmod f \ — & \cdots & — \ x^{n-1}\mathbf{a}_\ell \bmod f \ — & \\
\hline
— \ g \bmod f \ — & & & \\
\vdots & & & \\
— \ x^{n-1}g \bmod f \ — & & & \\
\hline
 & \ddots & & \\
\hline
 & & — \ g \bmod f \ — & \\
 & & \vdots & \\
 & & — \ x^{n-1}g \bmod f \ — &
\end{array}\right)$$

Example Lattice: What about for the Mersenne prime ring?

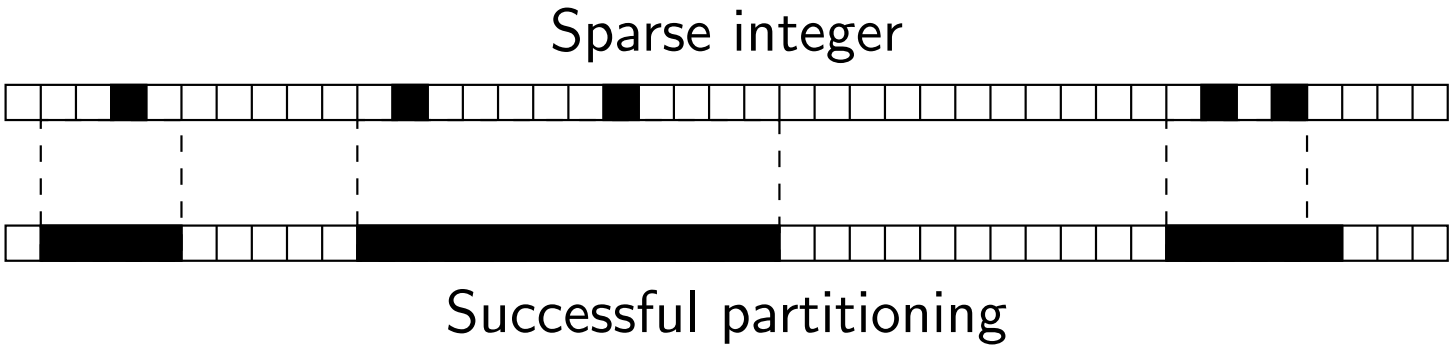
$$
\left(\begin{array}{c|c|c|c}
- \ \mathbf{b}_1 \ - & \cdots & - \ \mathbf{b}_\ell \ - & w \\
\hline
- \ \mathbf{a}_1 \ - & \cdots & - \ \mathbf{a}_\ell \ - & \\
\vdots & & \vdots & \\
- \ 2^{n-1}\mathbf{a}_1 \bmod p \ - & \cdots & - \ 2^{n-1}\mathbf{a}_\ell \bmod p \ - & \\
\hline
\begin{matrix} -2 & 1 & 0 & \cdots & 0 \\ & & \ddots & & \\ 1 & 0 & \cdots & 0 & -2 \end{matrix} & & & \\
\hline
 & \ddots & & \\
\hline
 & & \begin{matrix} -2 & 1 & 0 & \cdots & 0 \\ & & \ddots & & \\ 1 & 0 & \cdots & 0 & -2 \end{matrix} &
\end{array}\right)
$$

Example Lattice: What about for the Mersenne prime ring?

$$
\left(\begin{array}{c|c|c|c}
\multicolumn{2}{c|}{— \ \mathbf{b}_1 \ —} & \cdots & — \ \mathbf{b}_\ell \ — \quad w \\
\hline
\multicolumn{2}{c|}{\begin{array}{c} — \ \mathbf{a}_1 \ — \\ \vdots \\ — \ 2^{n-1}\mathbf{a}_1 \bmod p \ — \end{array}} & \cdots & \begin{array}{c} — \ \mathbf{a}_\ell \ — \\ \vdots \\ — \ 2^{n-1}\mathbf{a}_\ell \bmod p \ — \end{array} \\
\hline
\begin{array}{ccccc} -2 & 1 & 0 & \cdots & 0 \\ & \ddots & & & \\ 1 & 0 & \cdots & 0 & -2 \end{array} & & & \\
\hline
 & & \ddots & & \\
\hline
 & & & \begin{array}{ccccc} -2 & 1 & 0 & \cdots & 0 \\ & & \ddots & & \\ 1 & 0 & \cdots & 0 & -2 \end{array} & \\
\hline
\multicolumn{4}{l}{\left(\begin{array}{ccccc|c|c|c} -2 & 1 & 0 & \cdots & 0 & & & \end{array}\right.}
\end{array}\right)
$$

A Combinatorial Attack

- short vector we want has low hamming weight
- Idea:
  - partition $\{0, \ldots, n-1\}$ into consecutive active and inactive blocks
  - hope all 1s fall into active blocks



- Perform simple lattice reduction: recover secret if partition is correct
  - Unlikely to guess a correct partition
  - Attack is exponential in the hamming weight h

Example Lattice: The Mersenne prime ring

$$
\left(\begin{array}{c|c|c|c}
\multicolumn{2}{c|}{— \quad \mathbf{b}'_1 \quad —} & \cdots & — \quad \mathbf{b}'_\ell \quad — \qquad\qquad w' \\
\hline
\multicolumn{2}{c|}{— \quad \mathbf{a}'_1 \quad —} & \cdots & — \quad \mathbf{a}'_\ell \quad — \\
\multicolumn{2}{c|}{\vdots} & & \vdots \\
\multicolumn{2}{c|}{— \quad 2^{n-1}\mathbf{a}'_1 \bmod p \quad —} & \cdots & — \quad 2^{n-1}\mathbf{a}'_\ell \bmod p \quad —\\
\hline
\begin{matrix} -2^{\lambda_1} & 1 & 0 & \cdots & 0 \end{matrix} & & & \\
\begin{matrix} & & \ddots & & \end{matrix} & & & \\
\begin{matrix} 1 & 0 & \cdots & 0 & -2^{\lambda_k} \end{matrix} & & & \\
\hline
 & & \ddots & & \\
\hline
 & & \begin{matrix} -2^{\mu_1} & 1 & 0 & \cdots & 0 \end{matrix} & \\
 & & \begin{matrix} & & & \ddots & \end{matrix} & \\
 & & \begin{matrix} 1 & 0 & \cdots & 0 & -2^{\mu_j} \end{matrix} &
\end{array}\right)
$$

Example Lattice: The Mersenne prime ring

$$
\left(\begin{array}{c|c|c|c}
 & \cdots & & \\
— \ \mathbf{b}'_1 \ — & & — \ \mathbf{b}'_\ell \ — & w' \\
\hline
— \ \mathbf{a}'_1 \ — & \cdots & — \ \mathbf{a}'_\ell \ — & \\
\vdots & & \vdots & \\
— \ 2^{n-1}\mathbf{a}'_1 \bmod p \ — & \cdots & — \ 2^{n-1}\mathbf{a}'_\ell \bmod p \ — & \\
\hline
\begin{matrix} -2^{\lambda_1} & 1 & 0 & \cdots & 0 \\ & & \ddots & & \\ 1 & 0 & \cdots & 0 & -2^{\lambda_k} \end{matrix} & & & \\
\hline
 & \ddots & & \\
\hline
 & & \begin{matrix} -2^{\mu_1} & 1 & 0 & \cdots & 0 \\ & & & \ddots & \\ 1 & 0 & \cdots & 0 & -2^{\mu_j} \end{matrix} & \\
\end{array}\right)
$$

$$
\left(\begin{array}{c|c|c|c}
 & & & \\
— \ \mathbf{e}'_1 \ — & & — \ \mathbf{e}'_\ell \ — & w' \\
 & & & \\
\end{array}\right)
$$

The attacks: Why they work?

Standard LWE Family:

- q a *large integer* in comparison to the coefficients of small elements
- No small intrinsic vectors
  - strong lattice reduction works

Large Integer Arithmetic LWE Family:

- $x - 2$ has *small coefficients*
  - small elements are very sparse (lots of zeros)
- Guessing attacks work
  - avoids small intrinsic vectors

The middle ground

- Two extremes
  - large integer q
  - polynomial with small coefficients like $x - 2$
- Large middle ground for g
  - which attack is the most efficient?
  - do other attacks apply?
- Interesting research question
  - determine regions where each attack is most efficient
  - find new attacks?

A recipe for constructing (potentially) hard problems

1. Select the parent ring $R = \mathbb{Z}[x]/(f(x))$ by choosing f.
2. Select the error distribution on R.
3. Select the ciphertext modulus $g(x)$ subject to constraints.
4. Select the rank m of the module.
5. Select the hard problem family:

A recipe for constructing (potentially) hard problems

1. Select the parent ring $R = \mathbb{Z}[x]/(f(x))$ by choosing f.
2. Select the error distribution on R.
3. Select the ciphertext modulus $g(x)$ subject to constraints.
4. Select the rank m of the module.
5. Select the hard problem family:

   Ideal-LWE, Ideal-NTRU or Ideal-SIS

Problems using Ideal-LWE

$m = 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0		
	1		
	$\vdots$		

$m > 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0		
	1		
	$\vdots$		

Problems using Ideal-LWE

$m = 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0		
	1		
	$\vdots$		

$m > 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0	LWE	
	1		
	$\vdots$		

Problems using Ideal-LWE

$m = 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0		RLWE
	1		
	$\vdots$		

$m > 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0	LWE	
	1		
	$\vdots$		

Problems using Ideal-LWE

$m = 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0		RLWE
	1		
	$\vdots$		

$m > 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0	LWE	M-LWE
	1		
	$\vdots$		

Problems using Ideal-LWE

$m = 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0		RLWE
	1		I-RLWE, MLHC
	$\vdots$		

$m > 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0	LWE	M-LWE
	1		
	$\vdots$		

Problems using Ideal-LWE

$m = 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0		RLWE
	1		I-RLWE, MLHC
	$\vdots$		

$m > 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0	LWE	M-LWE
	1		I-MLWE (Three Bears)
	$\vdots$		

Problems using Ideal-LWE

$m = 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0	1-dimensional LWE	RLWE
	1		I-RLWE, MLHC
	$\vdots$		

$m > 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0	LWE, LPN, matrix LWE	M-LWE
	1		I-MLWE (Three Bears)
	$\vdots$		

Problems using Ideal-LWE

$m = 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0	1-dimensional LWE	RLWE
	1	$*$	I-RLWE, MLHC
	$\vdots$	$*$	

$m > 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0	LWE, LPN, matrix LWE	M-LWE
	1	$*$	I-MLWE (Three Bears)
	$\vdots$	$*$	

Problems using Ideal-LWE

$m = 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0	1-dimensional LWE	RLWE
	1	$*$	I-RLWE, MLHC
	$\vdots$	$*$	?

$m > 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0	LWE, LPN, matrix LWE	M-LWE
	1	$*$	I-MLWE (Three Bears)
	$\vdots$	$*$	?

Problems using Ideal-NTRU

$m = 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0	?	NTRU, NTRU Prime
	1	$*$	MLHR
	$\vdots$	$*$	?

$m > 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0	matrix NTRU	MaTRU
	1	$*$	?
	$\vdots$	$*$	?

Problems using Ideal-SIS

$m = 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0	modular SSP	Ring-SIS
	1	$*$	?
	$\vdots$	$*$	?

$m > 1$		$\deg(f) = 1$	$\deg(f) > 1$
$\deg(g)$	0	SIS	M-SIS
	1	$*$	?
	$\vdots$	$*$	?

Thank you!
Questions?