Parameter Selection in Ring-LWE-based Fully Homomorphic Encryption

Rachel Player

Information Security Group, Royal Holloway, University of London

based on joint works with

Martin R. Albrecht, Hao Chen, Kim Laine, Sam Scott, and Yuhou Xia

London-ish Lattice Coding & Crypto Meeting — September 29, 2017

Table of Contents

Setting the scene

Lattice-based crypto:

- Candidate for post-quantum crypto
- Parameter selection is a drawback

Setting the scene

Lattice-based crypto:

- Candidate for post-quantum crypto
- Parameter selection is a drawback

Fully Homomorphic Encryption:

Setting the scene

Lattice-based crypto:

- Candidate for post-quantum crypto
- Parameter selection is a drawback

Fully Homomorphic Encryption:

- the coolest application of lattice-based crypto

Setting the scene

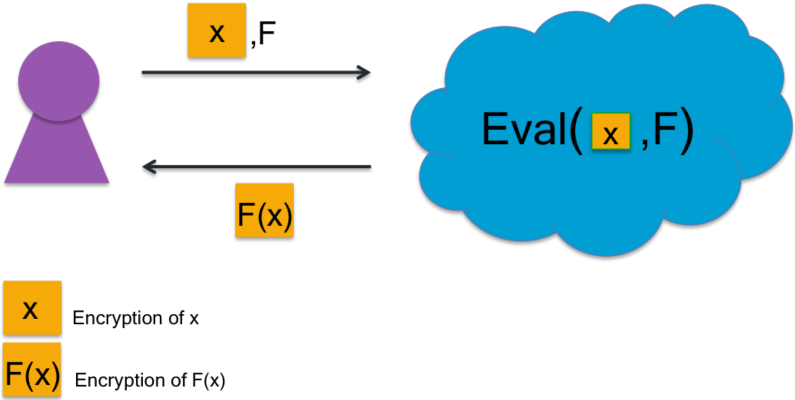
Lattice-based crypto:

- Candidate for post-quantum crypto
- Parameter selection is a drawback

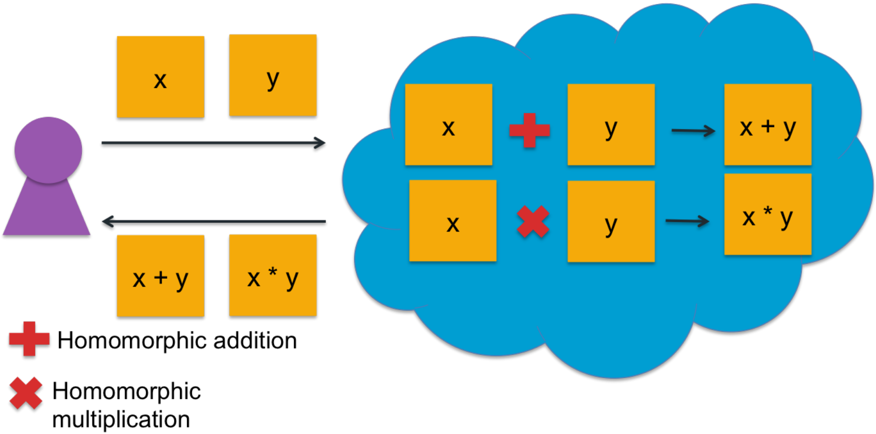
Fully Homomorphic Encryption:

- the coolest application of lattice-based crypto
- an interesting setting for parameter selection

What is homomorphic encryption?



Achieving homomorphic encryption



Applications of homomorphic encryption

- Healthcare
- Genomics
- Private set intersection
- Signal processing
- Machine learning
- ...

Is homomorphic encryption practical?

- First schemes very impractical

Is homomorphic encryption practical?

- First schemes very impractical
- Many implementations now exist:
  - HElib
  - SEAL
  - FV-NFLlib, Palisade, HEAAN, cuHE, TFHE, ...

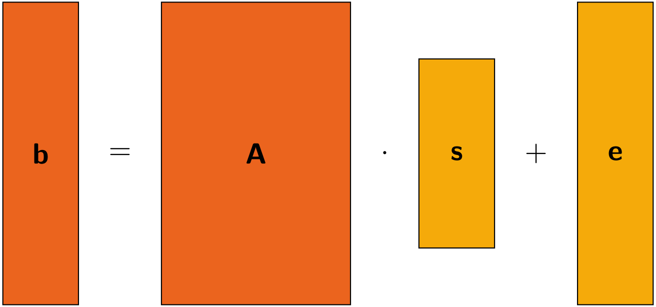
Is homomorphic encryption practical?

- First schemes very impractical
- Many implementations now exist:
  - HElib
  - SEAL
  - FV-NFLlib, Palisade, HEAAN, cuHE, TFHE, ...
- Standardisation effort: `https://homomorphicencryption.org`

Is homomorphic encryption practical?

- First schemes very impractical
- Many implementations now exist:
  - HElib
  - SEAL
  - FV-NFLlib, Palisade, HEAAN, cuHE, TFHE, ...
- Standardisation effort: `https://homomorphicencryption.org`
- Results for specific applications

Learning with Errors (LWE) [R05]



- Search: given $\mathbf{A}$ and $\mathbf{b}$, recover $\mathbf{s}$
- Decision: distinguish whether $(\mathbf{A}, \mathbf{b})$ is chosen as LWE or uniformly at random

Ring LWE definition

The ring R_q

Let n be a power of 2 and define

$$R_q = \mathbb{Z}_q[x]/(x^n + 1)$$

Ring LWE definition

The ring R_q

Let n be a power of 2 and define

$$R_q = \mathbb{Z}_q[x]/(x^n + 1)$$

Ring LWE (Decision)

Let $s \in R_q$ be a secret. Let $a \leftarrow R_q$ be chosen uniformly at random. Let χ be a distribution over R_q. Let $e \leftarrow \chi$. Distinguish $(a, b = as + e) \in R_q \times R_q$ from uniformly random $(a, b) \in R_q \times R_q$.

Why is n a power of two?

Theorem [LPR12]

There is a polynomial time quantum reduction from approximate SIVP (Shortest Independent Vector Problem) on ideal lattices in K to Decision Ring-LWE in R given a fixed number of samples, where the error distribution is a fixed spherical Gaussian over the field tensor product $K_{\mathbb{R}} = K \otimes_{\mathbb{Q}} \mathbb{R}$.

If $n = 2^k$:

- easy to implement
- performance benefit

What are the parameters?

A (Ring) LWE instance is specified by:

- n dimension
- q modulus
- α error distribution

where the standard deviation σ of χ satisfies

$$\sigma = \frac{\alpha q}{\sqrt{2\pi}}$$

Table of Contents

Is my Ring-LWE-based scheme secure?

- Parameters n, q, α in the scheme imply an underlying Ring LWE instance

- Treat Ring LWE instance as an LWE instance

- Observe that LWE instance is hard to solve

LWE based FHE parameters are atypical

Typical LWE parameters (Regev)

- q polynomial in n
- $\alpha q = \sqrt{n}$

LWE based FHE parameters are atypical

Typical LWE parameters (Regev)

- q polynomial in n
- $\alpha q = \sqrt{n}$

FHE parameters

- huge q
- tiny error distribution e.g. $\alpha q = 8$
- small secret $\|s\| = 1$
- possibly sparse secret

So how hard is (small secret) LWE, anyway?

Theory

LWE with binary secret in dimension $n \log q$ is as hard as general LWE in dimension n. [BLP+13,MP13]

- Many approaches for solving LWE
- Even more in the case of small and/or sparse secret

[APS15] estimator for hardness of LWE instances

https://bitbucket.org/malb/lwe-estimator

input LWE instance n, q, α

output estimates of runtime, memory, samples

Can optionally specify:

- Limited samples [BBGS17]
- Secret distribution
- Lattice reduction cost method

Running example: SEAL [DGBL+15,LP16,CLP16,CLP17]

- Homomorphic encryption library
- Developed by Microsoft Research
- Current version v2.2, June 2017
- Implements FV scheme [FV12]

`sealcrypto.org`

FV is IND-CPA secure if Ring LWE is hard

SecretKeyGen: Output $s \xleftarrow{\$} R_2$

PublicKeyGen: Sample $a \xleftarrow{\$} R_q$, and $e \leftarrow \chi$. Output

$$(p_0, p_1) = ([-(as + e)]_q, a)$$

Encrypt($(p_0, p_1), m$): Sample $u \xleftarrow{\$} R_2$, and $e_1, e_2 \leftarrow \chi$. Output

$$(c_0, c_1) = ([\Delta m + p_0 u + e_1]_q, [p_1 u + e_2]_q)$$

Choosing SEAL parameters for security

Already fixed are

- n a power of two
- $\sigma = 3.2$
- some threshold λ

Choosing SEAL parameters for security

Already fixed are

- n a power of two
- $\sigma = 3.2$
- some threshold λ

Find an acceptable bit length of q

- Choose initial bit length K
- Use [APS15] estimator to determine best attack for n, $q = 2^K$, $\alpha = 8/q$
- If best attack costs less than λ, decrement K and repeat
- If best attack costs more than λ, stop

Estimate of SEAL v2.2 security [CLP17]

n	q	α	usvp	dec	dual
2048	$2^{60} - 2^{14} + 1$	$8/q$	115.5	127.1	118.4
4096	$2^{116} - 2^{18} + 1$	$8/q$	119.7	125.3	121.2
8192	$2^{226} - 2^{26} + 1$	$8/q$	123.6	126.3	124.0
16384	$2^{435} - 2^{33} + 1$	$8/q$	129.5	130.7	130.2
32768	$2^{889} - 2^{54} - 2^{53} - 2^{52} + 1$	$8/q$	127.3	127.6	127.4

Table: Estimates of the cost of solving LWE instances underlying SEAL v2.2 default parameters. Obtained using commit cc5f6e8 of the estimator in [APS15].

Table of Contents

Noise and correctness

- FHE ciphertexts all have noise
- Noise grows with homomorphic operations
- If noise too large, decryption will fail

Noise and correctness

- FHE ciphertexts all have noise
- Noise grows with homomorphic operations
- If noise too large, decryption will fail

The better our understanding of noise the easier it is to choose parameters, and we may be able to choose smaller parameters

The FV scheme [FV12]

SecretKeyGen: Output $s \xleftarrow{\$} R_2$

PublicKeyGen(s): Sample $a \xleftarrow{\$} R_q$, and $e \leftarrow \chi$. Output

$$(p_0, p_1) = ([-(as + e)]_q, a)$$

Encrypt($(p_0, p_1), m$): Sample $u \xleftarrow{\$} R_2$, and $e_1, e_2 \leftarrow \chi$. Output

$$(c_0, c_1) = ([\Delta m + p_0 u + e_1]_q, [p_1 u + e_2]_q)$$

Decrypt(s, (c_0, c_1)): Output

$$\left[\left\lfloor \frac{t}{q}[c_0 + c_1 s]_q \right\rceil\right]_t$$

Existing notions of noise in FV

Inherent Noise [FV12,CLP16]

- The inherent noise is v_{inh} such that $[c_0 + c_1 s]_q = \Delta m + v_{inh}$.
- We require $\|v_{inh}\|_\infty < \frac{q}{2t} - \frac{t}{2}$

Critical quantity [CS16]

- The critical quantity is $v_{inh} - \frac{r_t(q)}{t} m$
- We require $\|v_{inh} - \frac{r_t(q)}{t} m\|_\infty < \frac{\Delta}{2}$

We want noise to be the thing which causes decryption to fail if it is too large

Recall FV Decryption: $m \mod t = \left[\left\lfloor \frac{t}{q} [c_0 + c_1 s]_q \right\rceil \right]_t$

Invariant noise [CLP17]

$$\frac{t}{q}(c_0 + c_1 s) = m + v + at$$

The norm $\|v\|_\infty$ is the invariant noise.

Invariant noise

By definition

$$\left\lfloor \frac{t}{q}[c_0 + c_1 s]_q \right\rceil = m + \lfloor v \rceil + a' t$$

So FV decryption succeeds

$$\left[\left\lfloor \frac{t}{q}[c_0 + c_1 s]_q \right\rceil \right]_t = m \mod t \qquad \text{if } \| v \|\|_\infty < \frac{1}{2}$$

Invariant noise

By definition

$$\left\lfloor \frac{t}{q}[c_0 + c_1 s]_q \right\rceil = m + \lfloor v \rceil + a' t$$

So FV decryption succeeds

$$\left[\left\lfloor \frac{t}{q}[c_0 + c_1 s]_q \right\rceil \right]_t = m \mod t \qquad \text{if } \|v\|\|_\infty < \frac{1}{2}$$

Noise budget

Initial noise in a fresh ciphertext is very small, and even in later ciphertexts we have $2\|v\|\|_\infty < 1$ if decryption succeeds. Easier to work with the noise budget defined as $-\log_2(2\|v\|)|_\infty$.

Homomorphic operations in SEAL

Addition $(\mathtt{ct}_0, \mathtt{ct}_1)$: Output

$$(\mathtt{ct}_0[0] + \mathtt{ct}_1[0], \mathtt{ct}_0[1] + \mathtt{ct}_1[1])$$

Multiplication $(\mathtt{ct}_0, \mathtt{ct}_1)$: Compute

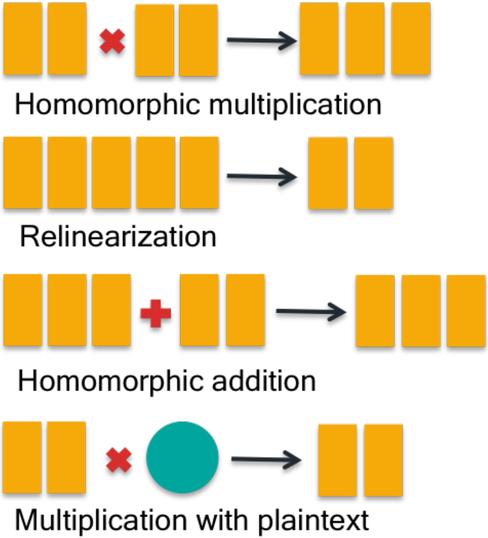
$$c_0 = \left[\left\lfloor \frac{t}{q} \mathtt{ct}_0[0]\mathtt{ct}_1[0] \right\rceil\right]_q$$

$$c_1 = \left[\left\lfloor \frac{t}{q} (\mathtt{ct}_0[0]\mathtt{ct}_1[1] + \mathtt{ct}_0[1]\mathtt{ct}_1[0]) \right\rceil\right]_q$$

$$c_2 = \left[\left\lfloor \frac{t}{q} \mathtt{ct}_0[1]\mathtt{ct}_1[1] \right\rceil\right]_q .$$

Output (c_0, c_1, c_2).

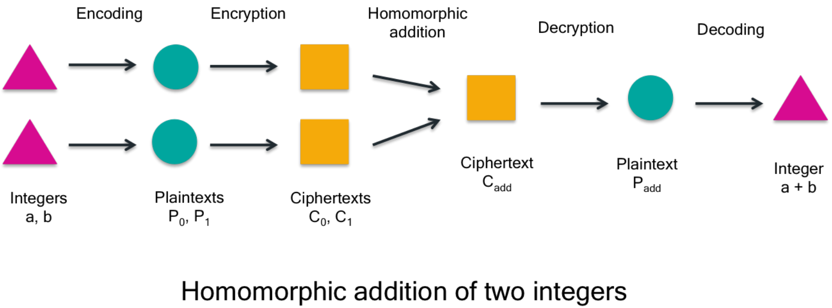
Example SEAL operations



Why is invariant noise better than inherent noise?

	Inherent noise bound	Invariant noise bound
Initial	$B(1+2\delta)$	$\frac{t}{q}\|m\| + \frac{t}{q}B(1+2\delta)$
Addition	$\|v_{1_{inh}}\| + \|v_{2_{inh}}\| + t$	$\|v_1\| + \|v_2\|$
Multiplication	$+\frac{\delta^{J+1}-1}{2(\delta-1)}$ $+\left(\delta t + \frac{\delta t(\delta^{j_2+1}-1)}{2(\delta-1)} + \frac{\delta t^2}{2q}\right)\|v_{1_{inh}}\|$ $+\left(\delta t + \frac{\delta t(\delta^{j_1+1}-1)}{2(\delta-1)} + \frac{\delta t^2}{2q}\right)\|v_{2_{inh}}\|$ $+\left(\frac{3\delta t}{q}\right)\|v_{1_{inh}}\| \cdot \|v_{2_{inh}}\|$ $\frac{2\delta t^2+t}{2} - \frac{3\delta t^3}{4q} + \frac{t^2\delta(\delta^{j_2+1}+\delta^{j_1+1}-2)}{4(\delta-1)}$	$\frac{t(\delta^{J+1}-1)}{2q(\delta-1)}$ $+\left(\delta t + \frac{\delta t(\delta^{j_2+1}-1)}{2(\delta-1)}\right)\|v_1\|$ $+\left(\delta t + \frac{\delta t(\delta^{j_1+1}-1)}{2(\delta-1)}\right)\|v_2\|$ $+3\delta \cdot \|v_1\| \cdot \|v_2\|$
Relinearization	$\|v_{inh}\| + (M-N)(\ell+1)\delta Bw$	$\|v\| + \frac{t}{q}(M-N)(\ell+1)\delta Bw$
Multiply plain	$\frac{\delta t}{2}\|v_{inh}\| + \frac{t}{2} + \frac{\delta t^2}{4}$	$\frac{\delta t}{2}\|v\|$
Add plain	$\|v_{inh}\| + t$	$\|v\| + \frac{t^2}{2q}$
Negation	$\|v_{inh}\| + t$	$\|v\|$
Subtraction	$\|v_{1_{inh}}\| + \|v_{2_{inh}}\| + t$	$\|v_1\| + \|v_2\|$

Encoding



In SEAL, plaintext space is $R_t = \mathbb{Z}_t[x]/(x^n + 1)$

SEAL automatic parameter selection

Input descriptions of:

- computation
- plaintext

SEAL automatic parameter selection

Input descriptions of:

- computation
- plaintext

The tool simulates noise growth and plaintext coefficient growth to find optimal parameters:

SEAL automatic parameter selection

Input descriptions of:

- computation
- plaintext

The tool simulates noise growth and plaintext coefficient growth to find optimal parameters:

- Sets error distribution as default

SEAL automatic parameter selection

Input descriptions of:

- computation
- plaintext

The tool simulates noise growth and plaintext coefficient growth to find optimal parameters:

- Sets error distribution as default
- Choose t as the smallest power of 2 such that decoding succeeds

SEAL automatic parameter selection

Input descriptions of:

- computation
- plaintext

The tool simulates noise growth and plaintext coefficient growth to find optimal parameters:

- Sets error distribution as default
- Choose t as the smallest power of 2 such that decoding succeeds
- Choose n and q from the default pairs as the smallest such that decryption succeeds

Table of Contents

Parameter selection for performance in SEAL

Choosing n (and σ)

- We are essentially done
- Power of two n turns out to be good for performance

Choosing t

- If t is such that $t | (q-1)$ then $r_t(q) = 1$

Parameter selection for performance in SEAL

n	q
8192	$2^{226} - 2^{26} + 1$
16384	$2^{435} - 2^{33} + 1$
32768	$2^{889} - 2^{54} - 2^{53} - 2^{52} + 1$

Choosing q

- Of the form $2^A - B$, where B is a small integer
- Of the form $2n|(q-1)$
  - In particular $4q \leq \beta$, where $\beta = 2^{64\lceil \log(q)/64 \rceil}$

Homomorphic operations in SEAL

Relinearization ($\mathtt{ct} = (c_0, c_1, c_2)$): Express c_2 in base w as

$$c_2 = \sum_{i=0}^{\ell} c_2^{(i)} w^i \, .$$

Set

$$c_0' = c_0 + \sum_{i=0}^{\ell} \mathtt{evk}[i][0] c_2^{(i)} \, ,$$

$$c_1' = c_1 + \sum_{i=0}^{\ell} \mathtt{evk}[i][1] c_2^{(i)} \, ,$$

and output (c_0', c_1').

Parameter selection for performance in SEAL

Relinearization: choosing w and ℓ

- Choice only affects relinearization and evaluation key generation
- Both relinearization and lack of relinearization can introduce noise
- Typical choice is $\log w = \frac{1}{2} \log q$
- Smaller $\log w$ is worse for performance
- Automatic parameter selection allows up to $\log w = \frac{1}{10} \log q$
- Essentially open problem to determine when to relinearize

Improved performance through new variant of FV [CLPX17]

- Plaintext modulus is $x - b$ rather than t [HS00]
- Plaintext space is $\mathbb{Z}/(b^n + 1)\mathbb{Z}$
- Easy encoding for integers and rationals
- Performs favourably compared to FV

The new scheme

Encoding m

For each $m \in \mathcal{M}$ denote by $\widehat{m}$ a shortest polynomial with $\|\widehat{m}\| \leq (b+1)/2$, such that $\widehat{m}(b) = m$ modulo $b^n + 1$

Encrypt($(p_0, p_1), m$): Sample $u \leftarrow \{-1, 0, 1\}$, and $e_0, e_1 \leftarrow \chi$. Output

$$(c_0, c_1) = \left(\Delta_b \widehat{m} + p_0 u + e_0, p_1 u + e_1\right) .$$

The new scheme

Encoding m

For each $m \in \mathcal{M}$ denote by $\widehat{m}$ a shortest polynomial with $\|\widehat{m}\| \leq (b+1)/2$, such that $\widehat{m}(b) = m$ modulo $b^n + 1$

Encrypt$((p_0, p_1), m)$: Sample $u \leftarrow \{-1, 0, 1\}$, and $e_0, e_1 \leftarrow \chi$. Output

$$(c_0, c_1) = (\Delta_b \widehat{m} + p_0 u + e_0, p_1 u + e_1) .$$

Decrypt$((c_0, c_1), s)$: Compute

$$\widehat{M} = \left\lfloor \frac{x-b}{q} [c_0 + c_1 s]_q \right\rceil .$$

Output $m' = \widehat{M}(b) \in \mathcal{M}$.

Comparison to FV

- Compare evaluation of regular circuit as in [CSVW16]
  - Do A additions and one multiplication, iterated D times
  - Inputs are integers of norm at most L

Comparison to FV

- Compare evaluation of regular circuit as in [CSVW16]
  - Do A additions and one multiplication, iterated D times
  - Inputs are integers of norm at most L
- Goal: Find (t, D) and (b, D) so D is maximised

Comparison to FV

- Compare evaluation of regular circuit as in [CSVW16]
  - Do A additions and one multiplication, iterated D times
  - Inputs are integers of norm at most L
- Goal: Find (t, D) and (b, D) so D is maximised

- Security for FV and new variant is the same so we can fix (n, q, σ)

Comparison to FV

- Compare evaluation of regular circuit as in [CSVW16]
  - Do A additions and one multiplication, iterated D times
  - Inputs are integers of norm at most L
- Goal: Find (t, D) and (b, D) so D is maximised

- Security for FV and new variant is the same so we can fix (n, q, σ)
- Noise and plaintext growth estimates give constraints

Encoders in FV

- Family parameterised by base B [DGBL+15] or Non-Adjacent Form
  - Small B enables smaller t
  - Large B enables shorter encodings

Encoders in FV

- Family parameterised by base B [DGBL+15] or Non-Adjacent Form
  - Small B enables smaller t
  - Large B enables shorter encodings

- Choose NAF since it outperforms $B = 2$ and $B = 3$ [CJLL17]

Results

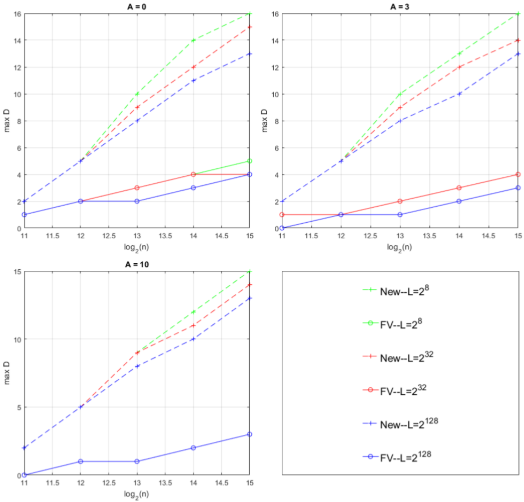

A = 0
A = 3
A = 10
max D
log_2(n)
New--L=2^8
FV--L=2^8
New--L=2^32
FV--L=2^32
New--L=2^128
FV--L=2^128

Table of Contents

More details

CLPX17 Hao Chen, Kim Laine, Rachel Player and Yuhou Xia. High-Precision Arithmetic in Homomorphic Encryption. ia.cr/2017/809

CLP17 Hao Chen, Kim Laine and Rachel Player. Simple Encrypted Arithmetic Library - SEAL (v2.2). *Technical report, 2017*. www.microsoft.com/en-us/research/publication/simple-encrypted-arithmetic-library-seal-v2-2/

APS15 Martin R. Albrecht, Rachel Player and Sam Scott. On the concrete hardness of Learning with Errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015. ia.cr/2015/046

Thank you! / Questions?

BLP+13 Z. Brakerski, A. Langlois, C. Peikert, O. Regev and D. Stehlé. Classical hardness of Learning with Errors. *In STOC*, 2013.

MP13 D. Micciancio and C. Peikert. Hardness of SIS and LWE with small parameters. *In Crypto*, 2013.

CJLL17 J. H. Cheon, J. Jeong, J. Lee and K. Lee. Privacy-preserving computations of predictive medical models with minimax approximation and Non-Adjacent Form. *In WAHC*, 2017.

CLP16 H. Chen, K. Laine and R. Player. Simple Encrypted Arithmetic Library - SEAL. *In WAHC, 2017*.

CS16 A. Costache and N. P. Smart. Which ring based somewhat homomorphic encryption scheme is best? *In CT-RSA*, 2016.

CSVW16 A Costache, N. P. Smart, S. Vivek and A. Waller. Fixed point arithmetic in SHE scheme. *In SAC*, 2016.

DGBL+15 N. Dowlin, R. Gilad-Bachrach, K. Laine, K. Lauter, M. Naehrig and J. Wernsing. Manual for using homomorphic encryption for bioinformatics. *Proceedings of the IEEE 105(3): 552–567*, 2017.

FV12 J. Fan and F. Vercauteren. Somewhat practical fully homomorphic encryption. *Eprint 2012/144*.

HS00 Jeffrey Hoffstein and Joseph Silverman. Optimizations for NTRU. *In Public Key Cryptography and Computational Number Theory*, 2001

LP16 K. Laine and R. Player. Simple Encrypted Arithmetic Library - SEAL (v2.0). *Technical report*, 2016.

LPR12 V. Lyubashevsky, C. Peikert and O. Regev. On ideal lattices and Learning with Errors over rings. *Eprint 2012/230 — Full version of paper appearing at Eurocrypt*, 2010.

R05 O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *In STOC*, 2005.

BBGS17 N. Bindel, J. Buchmann, F. Göpfert and M. Schmidt. Estimation of the hardness of the Learning with Errors problem with a restricted number of samples. *Eprint 2017/140*