Appendix A
Lattice Based Cryptography

Lattice ($\mathscr{L}(\mathscr{B})$) is a discrete additive subgroup of $\mathbb{R}^m$, which corresponds to the vector space generated by all linear combinations with integer coefficients of a set $\mathscr{B} = \{\overrightarrow{b_0} \ldots \overrightarrow{b_{n-1}}\}$, with $b_i \in R^m$. It can be represented as:

$$\mathscr{L}(\mathscr{B}) = \sum_{i=0}^{n-1} z_i \overrightarrow{b_0} : z_i \in \mathbb{Z} \tag{A.1}$$

Basic $\mathscr{B}$ can be represented as matrix B with vectors $\overrightarrow{b_i}$ as rows and Eq. A.2 can be represented as:

$$\mathscr{L}(\mathscr{B}) = \sum_{i=0}^{n-1} \overrightarrow{z} \times B : \overrightarrow{z_i} \in \mathbb{Z}^n \tag{A.2}$$

Few key features of lattice are:

- Lattices can have an infinite number of bases for $n \geq 2$.
- If two matrices B_1 and B_2 are associated with same lattice, they are related by an integer matrix U such that $|det(U)| = 1$ Hence, absolute value of determinants are same for all the bases of any lattice and denoted as $det(\mathscr{L})$.
- Every lattice base B has corresponding half open parallelepiped $\mathscr{P}(B) \leftarrow \sum_{i=0}^{n-1} z_i \overrightarrow{b_i} : z_i \in (\frac{-1}{2}, \frac{1}{2}]\}$
- Two vectors in a lattice are congruent ($\overrightarrow{x} = \overrightarrow{y}\,(\text{mod}(\mathscr{L}))$) if their difference is in the lattice ($\overrightarrow{x} - \overrightarrow{y}) \in \mathscr{L}$.
- The reduction of a vector y modulo a lattice base B, i.e ($\overrightarrow{x} = \overrightarrow{y}\,(\text{mod}(\mathscr{B}))$) corresponds to determining $\overrightarrow{x} \in \mathscr{P}(B)$, that can be computed as: $\overrightarrow{x} = \overrightarrow{y} - \lceil \overrightarrow{y} \times B^{-1} \rceil \times B = \lceil \overrightarrow{y} \times B^{-1} \rceil \times B$

To understand why lattice is suitable for cryptographic purpose, next we shall discuss few unique property and problems of lattice (Fig. A.1).

A. Chatterjee and K. M. M. Aung, *Fully Homomorphic Encryption in Real World Applications*, Computer Architecture and Design Methodologies,
https://doi.org/10.1007/978-981-13-6393-1

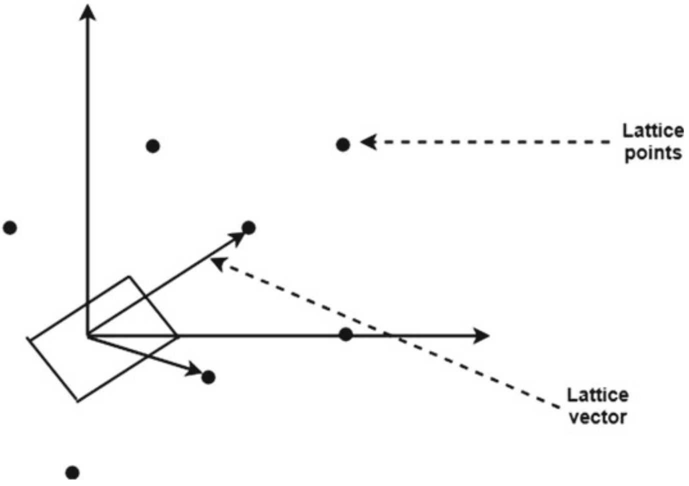

Fig. A.1 Lattice basics

Hermite Normal Form (HNF)

HNF base is unique to every lattice corresponds to a base H with following properties:

- $\forall i < j \; h_{i,j} = 0$
- $\forall j \; h_{j,j} > 0$
- $\forall i > j \; h_{i,j} \in (-h_{j,j}/2, h_{j,j}/2]$

HNF can be efficiently computed from any basis B of a lattice. Hence, this is considered a good choice for the public key of Lattice-Based Cryptosystems. Basic notion of security for Lattice-Based Cryptosystems are mostly dependent on the following:

- **Closest Vector Problem (CVP)**: Given a base $B \in \mathbb{R}^{n \times m}$ and $\overrightarrow{y} \in \mathbb{R}^m$, find $\overrightarrow{x} \in \mathscr{L}(B)$ such that $||\overrightarrow{y} - \overrightarrow{x}|| = min_{\overrightarrow{z} \in \mathscr{L}(B)}||\overrightarrow{y} - \overrightarrow{z}||$.
- **Shortest Vector Problem (SVP)**: Given a base $B \in \mathbb{R}^{n \times m}$ a, find $\overrightarrow{x} \in \mathscr{L}(B)$ such that $||\overrightarrow{x}|| = min_{\overrightarrow{z} \in \mathscr{L}(B)}||\overrightarrow{z}||$.
- **General Learning with Errors (GLWE)** (Brakerski et al. 2012): Let $n, \; m, \; q \; \in \mathbb{Z}$ and $R = \mathbb{Z}[t]/\phi_m(t)), R_q = R/qR$. Let χ be a Gaussian distribution over R. Given arbitrarily number of samples $(\overrightarrow{x_i}, y_i) \in R_q^{n+1}$, where $y_i = (\overrightarrow{x_i}, \; \overrightarrow{s}) + e_i$, where $\overrightarrow{x_i}, \; \overrightarrow{s} \leftarrow R_q^n$ sampled uniformly and $e_i \leftarrow \chi$, find $\overrightarrow{s}$.

Both LWE and lattices are connected in the following manner: Let us consider lattice $\mathscr{L}(B)$where the matrix $B \in \mathbb{Z}^{n \times t}$ has t numbers of $\overrightarrow{x_i}$ samples as columns. If closest vector $\overrightarrow{y'}$ can be computed to y with these samples that is equivalent to have a solution to LWE problem.

Reference

Brakerski Z, Gentry C, Vaikuntanathan V (2012) (leveled) Fully homomorphic encryption without bootstrapping. In: Innovations in theoretical computer science, pp 309–325

Appendix B
LWE Based FHE

Initial proposed FHE schemes are based on Gentry's seminal work with strong computational assumptions. These schemes are simplified with the learning with errors (LWE) security assumption (Regev et al. 2006). In the subsequent sections, we first mention the basics of LWE based schemes and discuss few basic constructions of LWE based FHE.

B.1 Basics of LWE Based Cryptosystem

LWE was first introduced by Regev et al. (2006) generalizing learning parity with noise problem. For positive integers n and $q \geq 2$, vector $s \in \mathbb{Z}_n^q$, probability distribution χ on $\mathbb{Z}_{||}^{\ltimes}$, let $A_{s,\chi}$ be the distribution obtained by choosing a random vector $a \leftarrow Z_q^n$ uniformly and a noise term $e \leftarrow \chi$ and outputting $(a, [(a, s) + e]_q) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. The decision version of LWE (DLWE) is to distinguish between noisy inner products and uniformly random samples from $\mathbb{Z}_q^n \times \mathbb{Z}_q$. It is defined as follows:

Definition B.1 For an integer $q = q(n)$ and an error distribution $\chi = \chi(n)$ over $\mathbb{Z}$, $DLWE_{n,m,q,\chi}$ is a problem to distinguish with non-negligible advantage, m samples chosen according to $A_{s,\chi}$ from m samples chosen from uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$. In this variant, the adversary gets oracle access to $A_{s,\chi}$.

It is interesting to note how this LWE problem can form the basic notion of security of cryptosystem. In general, LWE based crypto schemes generate ciphertexts Z_q for modulus q, which upon decryption produces noisy version of the message. Noise is added at encryption for security purposes. The decryption process is essentially computing an inner product of the ciphertext and the secret key vector. The noise should be below a certain threshold to retain the homomorphic property of the scheme. As discussed in Regev et al. (2006), parameters of LWE are mapped to any cryptosystem in the following way:

A. Chatterjee and K. M. M. Aung, *Fully Homomorphic Encryption in Real World Applications*, Computer Architecture and Design Methodologies,
https://doi.org/10.1007/978-981-13-6393-1

- **Secret Key**: Private key s is chosen as $s \in \mathbb{Z}_q^n$.
- **Public Key**: For $i = 1 \ldots m$, m vectors are chosen at random $a_1, \ldots a_m \in \mathbb{Z}_q^n$ uniformly from the distribution. $e_1, \ldots e_m \in \mathbb{Z}_q$ independently according to χ. Public key is considered as $(a_i, m_i)_{i=1}^m$ where $b_i = (a_i, s) + e_i$.
- **Encryption**: To encrypt a bit, random set S is chosen uniformly among 2^m subsets of $[m]$. The ciphertext (c) is:

$$c = \left(\sum_{i \in S} a_i, \sum_{i \in S} b_i\right), \textit{ for bit} = 0 \tag{B.1}$$

$$c = \left(\sum_{i \in S} a_i, \lfloor \frac{p}{2} \rfloor + \sum_{i \in S} b_i\right), \textit{ for bit} = 1 \tag{B.2}$$

- **Decryption**: Decryption of the ciphertext term with (a, b) pair is closer to 0 than $\lfloor \frac{p}{2} \rfloor$ if $b - (a, s)$ is 0, else 1.

B.2 Important LWE Based FHE Schemes

One of the notable scheme solely based on LWE assumption is explained in Brakerski et al. (2011), which shows a new simplified noise handling technique called relinearization supporting SHE scheme with much simpler hardness assumptions than ideals. Another important feature of this scheme is the formation of FHE from this SHE scheme without costly squashing or assumption of subset sum problem as detailed in Brakerski et al. (2011).

In this scheme, ciphertext $\overrightarrow{c}$ and secret key $\overrightarrow{s}$ are n dimensional vectors, where dot vectors of $(\overrightarrow{c}, \overrightarrow{s}) \approx \mu$ with small error that is removed by rounding. In this process, multiplication blows up the ciphertext size. Relinearization is a procedure that takes the long ciphertext that encrypts $\mu_1.\mu_2$ under a long key $\overrightarrow{s} \otimes \overrightarrow{s}$. This further compresses into a normal-sized n-dimensional ciphertext under a normal-sized n-dimensional key $\overrightarrow{s}$. This scheme is further improved by Craig Gentry, Amit Sahai and Brent Waters in their notable contribution in the paper (Gentry et al. 2013b). Now onwards we term that as GSW scheme.

Basics of GSW

- In this scheme, the requirement of relinearization has been removed using matrix multiplication using sub-cubic computation.
- This scheme proposes an identity-based FHE scheme, in which user with only the public parameters should be able to perform both encryption and homomorphism operations. The homomorphism operations should allow a user to take two ciphertexts encrypted to the same target identity, and homomorphically combine them to produce another ciphertext under the same target identity.
- These scheme can further be extended for a construction of homomorphic attribute-based encryption (ABE) with minor modifications.

References

Brakerski Z, Vaikuntanathan V (2011) Efficient fully homomorphic encryption from (Standard) LWE. FOCS 97–106

Gentry C, Sahai A, Waters B (2013b) Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster. Attribute-based. CRYPTO 75–92

Regev O (2006) Lattice-based cryptography. CRYPTO 131–141

Appendix C
GSW Based FHE Approach

Series of research contributions have been made in the direction of Gentry proposed homomorphic scheme (as discussed in BGV scheme Brakerski et al. 2012 and others) based on lattice based assumptions. In this contributions, circular security is one of the important assumption to obtain a FHE scheme from a leveled HE scheme. However, this is the main bottleneck in terms of performance while applying FHE in real world applications.

Another GSW based direction of research includes achieving the power of fully homomorphic computation in the simplest setting for bit-wise computation. Ducas et. al in their work (Ducas and Micciancio 2015) has proposed such FHE scheme where bootstrapping can be computed in less than seconds in bit level. In Table C.1, a comparison has been shown between such BGV and GSW based schemes. BGV based FHE construction details are discussed in chapter 2. In the subsequence section, we provide a brief idea of FHE schemes where bit-wise homomorphy is supported.

Table C.1 Present FHE trends in literature

	BGV based (Brakerski et al. 2012)	GSW based (Gentry et al. 2013a)
Performance	Slow in operation but processes huge number of bits	Faster operation but single bit processing
Operations	Limited set of operations due to parameter set:	Can support any arbitrary operation
Limitations	Operations can not be done 1. Comparison 2. Bit Extraction related operations 3. Addition etc	

A. Chatterjee and K. M. M. Aung, *Fully Homomorphic Encryption in Real World Applications*, Computer Architecture and Design Methodologies,
https://doi.org/10.1007/978-981-13-6393-1

Table C.2 Modulo 2 to Modulo 4 translation

m_1	m_2	$Dec[E(m_1 + m_2)]$ (modulo 2)	$Dec[E(m_1 \overline{\wedge} m_2)]$ (modulo 4)
$E(0)$	$E(0)$	0	1
$E(0)$	$E(1)$	1	0
$E(1)$	$E(0)$	1	0
$E(1)$	$E(1)$	0	0

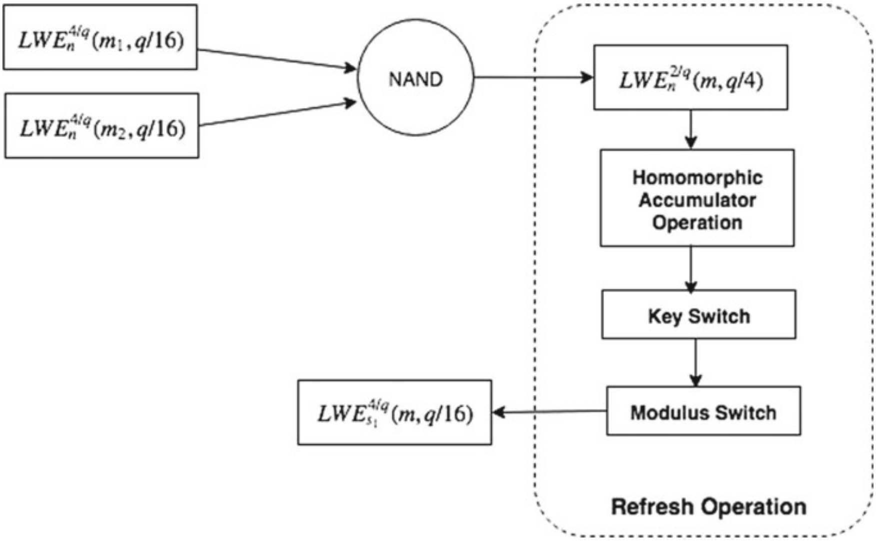

Fig. C.1 FHEW refreshing steps

C.1 Brief Details of Scheme Supporting Bit-Wise Homomorphy

Given two encrypted bits $E(b_1)$ and $E(b_2)$, this scheme aims to compute logical NAND of the two bits with following observation as shown in Table C.2. Computing $E(m_1 + m_2)$ in modulo 2 allows to homomorphically compute exclusive-or of two bits. From moving arithmetic modulo 2 to modulo 4, logical NAND computation is done during the bootstrapping process of this scheme. That indicates adding $E(m_1)$ and $E(m_2)$ generates $E(m)$, such that $E(m)$ of $m = 2$, if $m_1 \overline{\wedge} m_2 = 0$ or $m \in (0, 1)$, if $m_1 \overline{\wedge} m_2 = 1$.

This scheme homomorphically computes the NAND of two LWE encryptions. The noise introduced in this case is much lower, hence the refreshing technique is not so costly. The next step is the refreshing technique as shown in Fig. C.1. homomorphically evaluates $LWE_s^2(m, q/4)$ to $LWE_s^4(m, q/16)$. Details of the procedure can be found in the paper (Ducas and Micciancio 2015).

$$Refresh : LWE_s^2(m, q/4) \rightarrow LWE_s^4(m, q/16) \tag{C.1}$$

This scheme has been further enhanced in the paper (Chillotti et al. 2016a) where the bootstrapping has been improved from less than 1 s to less than 0.1 s. Some further improvements have been proposed in Chillotti et al. (2017a) with suitable packing techniques. These works form the mathematical background of TFHE library as discussed in the FHE library section.

References

Brakerski Z, Gentry C, Vaikuntanathan V (2012) (leveled) Fully homomorphic encryption without bootstrapping. In: Innovations in theoretical computer science, pp 309–325

Chillotti I, Gama N, Goubin L (2016a) Attacking FHE-based applications by software fault injections. IACR cryptology ePrint archive, vol 1164

Chillotti I, Gama N, Georgieva M, Izabachène M (2017a) Faster packed homomorphic operations and efficient circuit bootstrapping for TFHE. ASIACRYPT 377–408

Ducas L, Micciancio D (2015) FHEW: bootstrapping homomorphic encryption in less than a second. EUROCRYPT 617–640

Gentry C, Sahai A, Waters B (2013a) Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based. In: Advances in cryptology - CRYPTO 2013 - 33rd Annual cryptology conference, pp 75–92

Appendix D
FHE Based Libraries in Literature

Few FHE based libraries have been reported in literature with different features mentioned in Table D.1. In this section, we detail few of them:

Library HElib

HElib (Halevi et al. 2013b) is c++ based FHE software library based on Brakerski-Gentry-Vaikuntanathan (BGV) scheme (Brakerski et al. 2012) along with few optimizations. The library mostly focuses on the optimizations present in the paper (Gentry et al. 2012) and packing techniques from Smart and Vercauteren (2011). Work in Smart and Vercauteren (2011) presented a modified version of Gentry's fully homomorphic public key encryption scheme which supports SIMD style operations. This paper shows how to select parameters to enable such SIMD operations, whilst still maintaining practicality of the key generation technique of Gentry and Halevi. The proposed somewhat homomorphic scheme can be made fully homomorphic by recrypting all data elements seperately. However, this paper has shown a SIMD approach that can be used to perform recrypt in parallel supporting improved performance. this paper also demonstrates implementing AES homomorphically with this library.

Table D.1 FHE libraries in literature

Libraries	Scheme	Supporting libs
LibScarab (Perl et al. 2011)	SV (Smart and Vercauteren 2010)	GMP, FLINT MPFR, MPIR
HElib (Halevi et al. 2013b)	BGV (Brakerski et al. 2012)	NTL, GMP
FHEW (Ducas and Micciancio 2014)	FHEW (Ducas and Micciancio 2015)	FFTW
TFHE (Chillotti et al. 2017b)	TFHE	FFTW
SEAL (Laine et al. 2017)	FV12 (Fan and Vercauteren 2012)	No external dependency

A. Chatterjee and K. M. M. Aung, *Fully Homomorphic Encryption in Real World Applications*, Computer Architecture and Design Methodologies,
https://doi.org/10.1007/978-981-13-6393-1

Library FHEW

FHEW is open-source FHE library mathematically based on the paper "FHE bootstrapping in less than a second" (Ducas and Micciancio 2015). The name FHEW that is "Fastest Homomorphic Encryption in the West" is more of a reference to FFTW ("Fastest Fourier Transform in the West") than a claim about performance. In the paper, authors proposed method to homomorphically compute simple bit operations, and refresh (bootstrap) the resulting output within just about half a second in a consumer grade personal computer.

Most interesting feature of this library is that it provides bootstrapping in simplest possible setting which proved to be benificial later on in case of designing complex encrypted algorithms. To provide homomorphy, this library follows simple steps of encrypting single bits, and evaluating boolean NAND circuits on them. Basic idea is as follows: Given two encrypted bits $E(m_1)$ and $E(m_1)$, computing noisier version $E(m_1 + m_2)$ is very straight-forward. For arithmetic modulo 2, this is equivalent to compute exclusive-or of two bits. Next, logical NAND computation can be done by extending modulo 2 to modulo 4 with a bootstrapping (refreshing) technique.

Library TFHE

TFHE or "Fast Fully Homomorphic Encryption over the Torus" is another open-source library for FHE broadly a modification of FHEW. The mathematical background of this work is detailed in the paper "Faster fully homomorphic encryption: Bootstrapping in less than 0.1 s" (Chillotti et al. 2017b). This library supports the homomorphic evaluation of the 10 binary gates along with negation and encrypted multiplexer gate. Each binary gate takes about 13 ms single-core time to evaluate and the multiplexer gate takes about 26 CPU-ms. The gate-bootstrapping of TFHE has no restriction on the number of gates or on their composition that gives flexibility to realize real world design in encrypted form.

Library cuHE

Library cuHE or CUDA Homomorphic Encryption Library (cuHE Library 2018) is a GPU-accelerated library for HE schemes and homomorphic algorithms defined over polynomial rings. This library shows different techniques of memory minimization, memory and thread scheduling and low level CUDA related optimizations to take full advantage of the mass parallelism and high memory bandwidth of GPUs. This library is mostly SWHE with level limitations of HE.

References

Brakerski Z, Gentry C, Vaikuntanathan V (2012) (leveled) Fully homomorphic encryption without bootstrapping. In: Innovations in theoretical computer science, pp 309–325

Chillotti I, Gama N, Georgieva M, Izabachène M (2017b) TFHE: fast fully homomorphic encryption library over the torus. Retrieved from https://github.com/tfhe/tfhe (accessed September 2017)

cuHE. https://github.com/vernamlab/cuHE, Last Accessed: 11.10.2018

Ducas L, Micciancio D (2014) A fully homomorphic encryption library. Retrieved from https://github.com/lducas/FHEW (accessed October 2018)

Ducas L, Micciancio D (2015) FHEW: bootstrapping homomorphic encryption in less than a second. EUROCRYPT 617–640

Fan J, Vercauteren F (2012) Somewhat practical fully homomorphic encryption. Cryptology ePrint archive, Report 2012/144. Retrieved from http://eprint.iacr.org/2012/144

Gentry C, Halevi S, Smart NP (2012) Better bootstrapping in fully homomorphic encryption. In: Public key cryptography - PKC 2012 - 15th International conference on practice and theory in public key cryptography, pp 1–16

Halevi S, Shoup V (2013b) An implementation of homomorphic encryption. Retrieved from https://github.com/shaih/HElib (accessed September 2018)

Laine K, Chen H, Player R (2017) Simple encrypted arithmetic library. Retrieved from https://sealcrypto.codeplex.com/ (accessed September 2018)

Perl H, Brenner M, Smith M (2011) Poster: an implementation of the fully homomorphic smart-vercauteren crypto-system. In: ACM conference on computer and communications security, pp 837–840

Smart NP, Vercauteren F (2010) Fully homomorphic encryption with relatively small key and ciphertext sizes. In: Proceedings of the 13th international conference on practice and theory in public key cryptography, PKC'10, pp 420–443

Smart NP, Vercauteren F (2011) Fully homomorphic SIMD operations. IACR cryptology ePrint archive

Appendix E
Attacks on SWHE and FHE

In spite of the fact that HE specially FHE is considered to be the holy grail of cryptography, it is important to note that FHE is also susceptible to different kinds of attacks. Following the discussion of Martins et al. (2018), here we mention few important attacks against SWHE and FHE. In the next sections, the attacks are largely classified in two types: Passive and Active attacks.

E.1 Passive Attacks Against HE

In this section, we start our discussion about attacks against some known SWHE schemes like RSA and El-Gamal.

- **Attacks against RSA**: The notion of RSA security lies on the hardness of factoring large integers. However, referring to RSA discussion in Chap. 2, an attacker can find the secret key d by factoring modulus q since e is known. In the work (Boneh et al. 1999), brute-force attack on RSA has been shown to obtain the secret key. Most known and efficient algorithm about integer factorization is Pollard's General Number Field Sieve (GNFS) (Lenstra et al. 1990). Table E.1 mentions few notable contributions in this direction.
- **Attack against El-gamal cryptosystem**: Compared to integer factorization problem, large amount of research has been done in the direction of solving discrete logarithm problem which is the main notion of security for El-gamal Cryptosystem. Few notable contributions in this direction are highlighted in Joux et al. (2014), Menezes et al. (1996).
- **Attack against Lattice based cryptosystem (LBC)**: Lattice based cryptography is mostly dependent on two mathematical problems CVP and SVP. Overview of CVP and SVP are given in previous discussions. Few Attacks against LBC are mentioned in Table E.2 mostly against SVP and CVP problems.

A. Chatterjee and K. M. M. Aung, *Fully Homomorphic Encryption in Real World Applications*, Computer Architecture and Design Methodologies,
https://doi.org/10.1007/978-981-13-6393-1

Table E.1 Different passive attacks

Algorithm	Attack
Integer factorization	Pollard's general number field sieve (Lenstra et al. 1990)
Discrete logarithm problem	Shor algorithm and others (Joux et al. 2014;Shor 1990)
Lattice based attacks	
NTRU and variant of Gentry's scheme	Exploitation of ring structure leads to subexponential and quantum polynomial attacks (Albrecht et al. 2016; Cramer et al. 2016)
SVP	SVP solvers (Kuo et al. 2016)
CVP, SVP, RSA, AGCD problem	Lattice basis reduction algorithms (Lenstra et al. 1982)

Table E.2 Few active attacks against HE

Scheme	Attacks
Schemes with either HE additions or multiplications	IND-CCA1 secured but not IND-CCA2 (Lipmaa et al. 2008)
AGCD-, LWE-, and NTRU-based schemes	key recovery attacks [adversary is capable of getting the secret key with a polynomial amount of queries to a decryption oracle]. (Dahab et al. 2015)
Variant of SV SHE scheme in	IND-CCA1 secured but not IND-CCA2. (Loftus et al. 2010)

E.2 Active Attacks Against HE

All the attacks mentioned in the previous section refer to passive attacks which does not require any direct interference of the attacker with the target system. On the other hand, active attacks are based on the simple assumption that: If an attacker can identify which of two possible plaintexts between m_0 and m_1 can encrypt c with a probability more than 0.5, then the system is considered to be insecure. This type of security requirements can be conceptualized with few indistinguishably assumptions like:

- **Chosen plaintext attack** or **IND-CPA**
- **chosen ciphertext attack** or **IND-CCA1**
- **adaptive chosen ciphertext attack** or **IND-CCA2**

Semantic security assumptions are nicely explained in Loftus et al. (2011) by defining game between a challenger and an adversary A. For FHE, an attacker can decrypt arbitrary ciphertexts and the secret key is made public in an encrypted form.

Hence, FHE is not considered as IND-CCA1 and IND-CCA2 secured but IND-CPA secured. However, some of the SWHE schemes has proved to be IND-CCA1 secured (Loftus et al. 2011). Finally, in the paper (Chillotti et al. 2016a) authors have shown how FHE based applications can be susceptible to software fault injection attacks due to its ability to compute function.

References

Albrecht M, Bai S, Ducas L (2016) A subfield lattice attack on overstretched NTRU assumptions. Springer, Berlin, pp 153–178

Boneh D (1999) Twenty years of attacks on the RSA cryptosystem. Not AMS 46:203–213

Chillotti I, Gama N, Goubin L (2016a) Attacking FHE-based applications by software fault injections. IACR cryptology ePrint archive, vol 1164

Cramer R, Ducas L, Peikert C, Regev O (2016) Recovering short generators of principal ideals in cyclotomic rings. Springer, Berlin, pp 559–585

Dahab R, Galbraith S, Morais E (2015) Adaptive key recovery attacks on NTRU-based somewhat homomorphic encryption schemes. Cryptology ePrint archive, Report 2015/127. Retrieved from http://eprint.iacr.org/

Joux A, Odlyzko A, Pierrot C (2014) The past, evolving present, and future of the discrete logarithm. In: Open problems in mathematics and computational science. Springer International Publishing, Cham, pp 5–36

Kuo P-C, Schneider M, Dagdelen Ö, Reichelt J, Buchmann J, Cheng C-M, Yang B-Y (2011) Extreme enumeration on GPU and in clouds: how many dollars you need to break SVP challenges. In: Proceedings of the 13th international conference on cryptographic hardware and embedded systems (CHES)

Lenstra AK, Lenstra HW Jr, Lovsz L (1982) Factoring polynomials with rational coefficients. Math Ann 261:515–534

Lenstra AK, Lenstra HW Jr, Manasse MS, Pollard JM (1990) The number field sieve. In: STOC, pp 564–572

Lipmaa H (2008) On the CCA1-security of elgamal and Damgård's elgamal. IACR cryptology ePrint archive, vol 234

Loftus J, May A, Smart NP, Vercauteren F (2010) On CCA-secure fully homomorphic encryption. Cryptology ePrint archive, Report 2010/560. Retrieved from http://eprint.iacr.org/

Loftus J, May A, Smart NP, Vercauteren F (2011) On CCA-secure somewhat homomorphic encryption. Selected areas in cryptography, pp 55–72

Martins P, Sousa L, Mariano A (2018) A survey on fully homomorphic encryption: an engineering perspective. ACM Comput Surv 50(6):83:1–83:33

Menezes AJ, Vanstone SA, Van Oorschot PC (1996) Handbook of applied cryptography. CRC Press, Boca Raton

Shor PW (1994) Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings of the 35th annual symposium on foundations of computer science, pp 124–134 (1990)

Appendix F
Examples of Homomorphic Real World Applications

In this final section, we mention (in Table F.1 and in Table F.2) few real world problems where power of homorphic encrypted computation is extensively used. Design of these encrypted applications rightly justify the motivation of our book which discusses the steps and challenges of realizing traditional algorithms to their encrypted counterpart.

Table F.1 FHE real world applications

Domain	Application
Medical	Encrypted cardiac risk factor algorithm (Carpov et al. 2016)
Medical	Encrypted predictive analysis on medical data (Bos et al. 2014)
Medical	Long-term patient monitoring via cloud-based ECG data acquisition and encrypted analytics design (Kocabas et al. 2014)
Analytics [useful for business and medical]	Logistic regression over encrypted data (Chen et al. 2018)
Analytics	Big data analytics over encrypted datasets with seabed (Papadimitriou et al. 2016)
Analytics	Encrypted classification in machine learning (Graepel et al. 2012)
Data analysis	Encrypted statistical analysis (Lu et al. 2012)
Deep learning	Running convolutional neural networks (CNN) on encrypted data (Badawi et al. 2018)

A. Chatterjee and K. M. M. Aung, *Fully Homomorphic Encryption in Real World Applications*, Computer Architecture and Design Methodologies,
https://doi.org/10.1007/978-981-13-6393-1

Table F.2 FHE real world applications

Domain	Application
Financial	Encrypted financial computational model for cloud framework (Peng et al. 2016)
Cyber Physical System (CPS)	Primitives for computations on encrypted data for CPS systems (Hu et al. 2016)
Cyber Physical System (CPS)	Encrypting controller using FHE for security of cyber-physical systems (Kim et al. 2016)
Others	Secure distributed incremental information aggregation for smart grids using HE (Alabdulatif et al. 2017)
Others	Secure friend discovery in social strength-aware Proximity-Based Mobile Social Networks (PMSNs) (Niu et al. 2015)

References

Alabdulatif A, Kumarage H, Khalil I, Atiquzzaman M, Yi X (2017) Privacy-preserving cloud-based billing with lightweight homomorphic encryption for sensor-enabled smart grid infrastructure. IET Wirel Sens Syst 7(6):182–190

Badawi AA, Chao J, Lin J, Mun CF, Jie SJ, Tan BHM, Nan X, Aung KMM, Chandrasekhar VR (2018) The AlexNet moment for homomorphic encryption: HCNN, the first homomorphic CNN on encrypted data with GPUs. IACR cryptology ePrint archive, vol 1056

Bos JW, Lauter KE, Naehrig M (2014) Private predictive analysis on encrypted medical data. J Biomed Inform 50:234–243

Carpov S, Nguyen TH, Sirdey R, Costantino G, Martinelli F (2016) Practical privacy-preserving medical diagnosis using homomorphic encryption. CLOUD 593–599

Chen H, Gilad-Bachrach R, Han K, Huang Z, Jalali A, Laine K, Lauter KE (2018) Logistic regression over encrypted data from fully homomorphic encryption. IACR cryptology ePrint archive, vol 462

Graepel T, Lauter KE, Naehrig M (2012) ML confidential: machine learning on encrypted data. ICISC 1–21

Hu P, Mukherjee T, Valliappan A, Radziszowski S (2016) Evaluation of homomorphic primitives for computations on encrypted data for CPS systems. In: Smart city security and privacy workshop (SCSP-W), pp 1–5

Kim J, Lee C, Shim H, Cheon JH, Kim A, Kim M, Song Y (2016) Encrypting controller using fully homomorphic encryption for security of cyber-physical systems. IFAC-PapersOnLine 49(22):175–180

Kocabas O, Soyata T (2014) Private predictive analysis on encrypted medical data. J Biomed Inform 50:234–243. https://doi.org/10.4018/978-1-4666-5864-6.ch019

Lu W, Kawasaki S, Sakuma J (2017) Using fully homomorphic encryption for statistical analysis of categorical, ordinal and numerical data, NDSS

Niu B, He Y, Li F, Li H (2015) Achieving secure friend discovery in social strength-aware PMSNs. MILCOM 947–953

Papadimitriou A, Bhagwan R, Chandran N, Ramjee R, Haeberlen A, Singh H, Modi A, Badrinarayanan S (2016) Big data analytics over encrypted datasets with seabed. OSDI 587–602

Peng H-T, Hsu WWY, Ho J-M, Yu M-R (2016) Homomorphic encryption application on FinancialCloud framework. SSCI 1–5

Bibliography

Chillotti I, Gama N, Georgieva M, Izabachène M (2016b) Faster fully homomorphic encryption: bootstrapping in less than 0.1 seconds. In: ASIACRYPT (1), pp 3–33

UniCrypt. https://github.com/bfhevg/unicrypt/blob/master/README.md, Last Accessed: 11.10.2018