

Systematic literature review on bootstrapping procedure of FHE schemes

Sedat Akleylek [†]
Department of Computer Engineering
Istinye University
Istanbul 34010
Turkey

and

Institute of Computer Science
University of Tartu
Tartu 51009
Estonia

Adnan Karataş ^{*}
Department of Mathematics
Faculty of Sciences
Pamukkale University
Denizli 20160
Turkey

Abstract

Fully homomorphic encryption is one of the ultimate goals of the cryptography. It is the utmost solution for data privacy. The first seeds of the concept were planted nearly fifty-five years ago. After the thesis of Gentry, first theoretic structure was ready for the researchers. In the following years many different FHE schemes were suggested, and they were vastly improved. FHE schemes have noise or error added to the system in order to make the schemes secure, actually post quantumly secure in some cases. But this noise should be arranged regularly, not to be preponderated over the data. The core mechanism of the FHE is bootstrapping, which regulates the noise and enables us to do homomorphic operations without losing the decryption of plaintext. In this study, we collected research articles from reputable databases and reviewed them. By doing so, we make it easier to keep up with the latest advancements in bootstrapping of FHE schemes. Our goal is to provide a valuable

[†] E-mail: akleylek@gmail.com

^{*} E-mail: adnank@pau.edu.tr (Corresponding Author)

resource for anyone with background seeking to stay up-to-date on the latest developments in the bootstrapping procedure of FHE schemes. Also, we sought answers to our research questions. Lastly, we gave open problems for related studies.

Subject Classification: [2010] Primary 68P25, Secondary 11T71.

Keywords: Fully homomorphic encryption, Homomorphic encryption, Bootstrapping.

1. Introduction

The first ideas of homomorphic encryption appeared in scientific literature in 1978, were called privacy homomorphism [56]. After 30 years, Gentry's PhD thesis appeared [25] and showed that with bootstrapping, it is possible to make theoretically infinite number of calculations on encrypted data without decrypting it. The impact of the fully homomorphic encryption (FHE) on academic studies and different industrial areas is still growing.

We have to mention that, there are schemes one can make addition or multiplication on encrypted data such as [50]. In these schemes, the number of operations on ciphertexts is limited, and they are called (PHE) partial homomorphic encryption schemes. Later, somewhat homomorphic encryption (SHE) schemes are defined like. In these schemes, two different operations can be made until certain degree. Even though, there are boundaries on the number of operations, SHE schemes have usage areas for private database queries, calculation of statistical data such as regression and e-voting.

The fully homomorphic encryption means that anyone (not necessarily the keyholder) can compute the value of $f(\pi_1, \pi_2, \dots, \pi_i)$, as long as the function f can be efficiently calculated where $\pi_1, \pi_2, \dots, \pi_i$ are ciphertext. In addition, there should be no leakage of information whatsoever. From the very definition, it is no surprise that FHE is the ultimate goal of privacy and cryptography. Also, it is no coincidence that there are implementations of FHE in cloud computing, machine learning, storing of private data such as medical and financial records.

As we mentioned earlier, the bootstrapping technique made it possible to do the infinite number of operations on encrypted data. Since then, different FHE schemes have been defined, tested and upgraded. We will mention different FHE branches namely; ideal lattices branch, AGCD (approximate - greatest common divisor) branch, NTRU branch, (R)LWE branch.

We will remind to the reader basic concepts of the FHE schemes according to their branches in Section 2. Next, we will study the latest improvements on FHE schemes especially their bootstrapping procedure in Section 3. To make our study more reliable and trustworthy, we use systematic literature review technique which we give details in the mentioned section. In Section 4, we answer our research questions. Finally, we give open problems and conclusion in the Section 5.

2. Bootstrapping on some FHE schemes

In fully homomorphic systems, there is a noise in ciphertext which makes the scheme secure. However, this noise grows after each operation. If the noise outgrows, then it disrupts the plaintext, making it incoherent which makes the data unreadable. So, we need the bootstrapping technique in order to lower the noise to an acceptable level. There are two different approaches to bootstrapping, first one chooses to bootstrap after every operation, the other one fixes a noise level, after that threshold, the system bootstraps.

In Figure 1, m is a message and c is a ciphertext of m with public key k_1 . After homomorphic operations, f and g , the noise level of the ciphertext reaches the threshold. From this point, if more operations done to the ciphertext, the message will be unreadable. In order to prevent this problem, bootstrapping is used to lower the noise of ciphertext. After the bootstrapping, additional computations can be made. Note that, ciphertext can be stored, and all other homomorphic operations can be made by anyone without leakage of information about the message or private key. This fact is the core idea of fully homomorphic encryption.

In the following subsections, starting with FHE scheme on ideal lattices which is the first FHE scheme defined, we want to briefly mention FHE scheme branches.

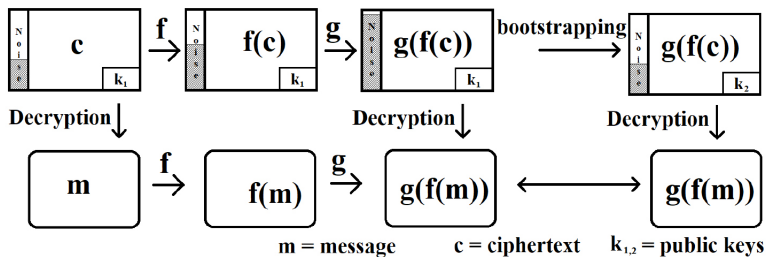


Figure 1

General structure of the bootstrapping process

2.1 FHE schemes based on ideal lattices

Ideal lattices have two mathematical properties, they are both lattice and ideal which are used to define cryptosystem with operations [59]. Gentry builds the idea similarly to server-aided cryptography and published the first bootstrapping technique in [25].

The author uses the sublattice $L(I) \subseteq \mathbb{Z}^m$ which is integer sublattice defined with the ideal I in his scheme. Then he uses the idea to express the same space with good (more orthogonal) and bad (skewed) basis for the mentioned lattice. Good basis is used as a secret key and bad basis is used as a public key, both of them can be used to encrypt the plaintext, but it is hard to obtain any information using bad basis. The security of the scheme depends on small principal ideal problem, the closest vector problem or polynomial coset problem depending on the version of FHE scheme. Newer and developed FHE schemes are using different hard problems in order to obtain security. The ideal lattice branch is not necessarily studied lately, but it is the first FHE scheme.

As given in the reference [25], the scheme is not bootstrappable because of the complexity of the decryption process. In order to overcome the issue, the author uses the squashing method to reduce the complexity of the decryption function where evaluation is possible homomorphically. To achieve such a goal, extra information about sk is embedded into the evaluation key.

The number one disadvantage of the scheme is that it uses mathematical structures with inefficient implementation. In order to clarify the problem it is enough to mention that implementation of the scheme has pk with size of 2.3 GB and re-encryption process takes 30 minutes [26]. Also, in 2016, a security issue with the scheme was shown by authors in [21] which makes the ideal lattice branch less appealing.

2.2 FHE schemes based on AGCD

The approximate - greatest common divisor problem is defined and studied in [34]. AGCD and sparse subset sum problems are used in definition of FHE scheme in 2010, [22].

The scheme was called as DGHV and assumes circular security, which means it is secure to encrypt secret key. For more information on circular security and key-dependent message security, see [37].

The DGHV scheme has a simple construction. It uses integers and relies on finding the common near divisor of p . One of the early implementations of the DGHV was [19]. In the work, it can be seen that the

scheme has obvious disadvantages in comparison to newly defined FHE schemes. The main one is its large public key size nearly 1 GB and other one is duration of re-encryption time approximately 15 minutes. The scheme receives new approaches and new techniques such as batching, scaling, modulo switching. For additional information, see [52].

One setback of the scheme is AGCD problem is quantumly broken by Shor algorithm [58], but the newer techniques and improvements in this branch or intertwinments of branches made integer based FHE to attract researchers from academia and industry [52].

2.3 FHE schemes based on NTRU

NTRU schemes rooted from the paper of the authors in 1998 [33]. The first NTRU based FHE scheme defined in 2012 [44]. Authors proposed the scheme to be used on-the-fly multiparty computation. The LTV scheme is using RLWE as a hard problem with DSPR which is Decisional Small Polynomial Ratio problem.

Yet another FHE scheme based on LTV is YASHE which was proposed in 2013. This scheme has two different variants. The first variant has same hard problems with LTV, RLWE and DSPR. The second one has only RLWE removing the DSPR in exchange for more complex key switching method [10].

Nowadays, NTRU based schemes still attract the researchers. Some of lately studies are.

2.4 FHE schemes based on (R)LWE

LWE is a hard problem introduced by Oded Regev in 2005 [55]. The author won the Gödel prize for this study. We skip the underlying mathematics of the learning with error problem and give some of FHE schemes which uses variants of it.

There are many schemes using LWE and its variants. The most known of the two are TFHE and CKKS [17]. But, BGV and GSW schemes are the earlier FHE schemes based on LWE [27]. There are lots of studies in this branch which we cannot fully cover. The popularity of the TFHE and CKKS arise from their strong code libraries and functional bootstrapping of TFHE and approximate calculation nature of the CKKS are unique properties of the schemes.

All the main FHE schemes are improved in different aspects. Our main focus is the bootstrapping mechanism and improvements of it. We

will state very specific studies from the literature and analyze them in the next section according to a systematic literature review process.

3.Systematic Literature Review on Bootstrapping

3.1 Methodology and data

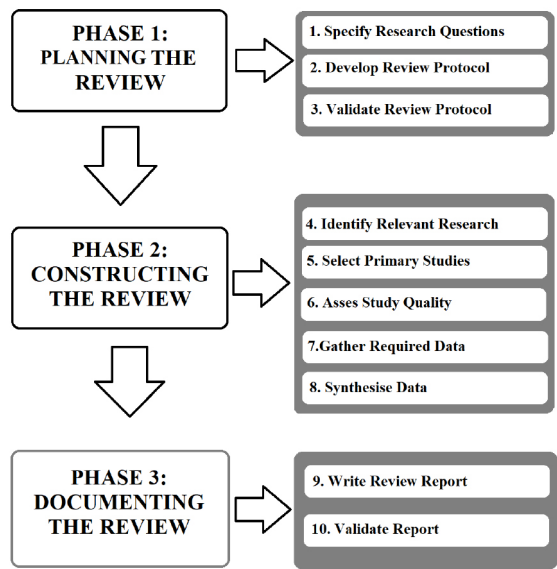


Figure 2
The SLR process

We approached this systematic literature review study with bibliometric analysis. This analysis technique is transparent, systematic and credible which is why it is widely used in the literature. This method reveals the trends, patterns of studies and overall understanding for the focused area.

For the SLR on bootstrapping we give general path in Figure 2. First, we plan the review, then build the general structure of the review. Furthermore, we give the manuscript its final form and report the findings of the study.

To fulfill this outline, we specify our research questions in Table 1.

Table 1
Research questions

	Research Questions
Q1	Which bootstrapping methodologies used in the FHE schemes ?
Q2	What kind of updates are made to bootstrapping process of the FHE schemes ?
Q3	Which algebraic methods used in the bootstrapping process ?
Q4	What kind of key approaches used in the FHE schemes?

For the process of the search, we use search combinations on title, abstract and keyword areas in the following databases.

- ScienceDirect, is a database by Elsevier which indexes scientific articles and books.
- Scopus, is a database that offers detailed search options on wide scientific literature.
- Web of Science, is a database which uses respected science citation index and scans more than 9000 scientific journals.
- IEEE Xplore, is a database dedicated to scientific and technical content published by the IEEE.

As an inclusion and exclusion criteria, we only focused on original research articles in English. We restrict our search to be between in 2018 and 2023 to attain latest published manuscripts. We use ‘fully homomorphic encryption’ or ‘homomorphic encryption’ and ‘bootstrapping’ terms in title, abstract and keywords areas to narrow the results. To be precise, we list search words for every database we mentioned earlier in Table 2. In addition, we excluded subject areas such as, art and humanities, social sciences, medicine, etc. Note that, we used similar search words in Google scholar and obtained more than 42000 works. After sorting the outcomes according to relevancy, we had to omit the results because of the quality of the studies.

Apart from original research papers, we want to mention review papers from the databases. The study, [6], investigates the bootstrapping process of the schemes, between 2009 and 2016. In the other surveys, like [2], the authors present general information about the homomorphic encryption, outsourced computation, applications for the HE and provide security information about the topic. Even though they give minimal

information about the inner procedures for the bootstrapping, the manuscripts have valuable information for those who seek to learn the general aspects of the homomorphic encryption and its applications. The difference of our survey from the mentioned ones is that, we focused on the bootstrapping process as we present details for researchers who want an overview about improvements in the bootstrapping via original research articles from respected databases.

In the following table we give our search words explicitly.

Table 2
Search words and results according to databases

Database	Search Parameters	Search Words	Results
Science Direct	Title, abstract, keywords	((“fully homomorphic encryption” OR “fhe” OR “homomorphic encryption” OR “he”) AND (“bootstrapping”))	5
Scopus	Title, abstract, keywords	TITLE-ABS-KEY(((fully AND homomorphic AND encryption) OR (homomorphic AND encryption) OR (he) OR (fhe)) AND (bootstrapping))	153
Web of Science	Title, abstract, keywords	((“fully homomorphic encryption” OR “fhe” OR “homomorphic encryption” OR “he”) AND (“bootstrapping”))	87
IEEE Xplore	All	((“fully homomorphic encryption” OR “fhe” OR “homomorphic encryption” OR “he”) AND (“bootstrapping”))	29

In order to obtain papers with improvements on bootstrapping procedure, we assessed and selected the manuscripts one by one. Consequently, we obtained 14 studies for this review. The overall process is depicted in the following Figure 3.

To sum up, we searched bootstrapping articles in 4 major databases. To observe the latest developments on homomorphic encryption schemes, we restricted our search to include only the last six years. To gather significant studies, we limited the options to contain just original research articles without irrelevant subject areas. After reading all manuscripts we excluded articles that do not contain bootstrapping improvements. So, we gathered 14 manuscripts which we indicate in Table 3. The table, so the manuscripts in it, are listed according to their respective FHE schemes and the time they are published.

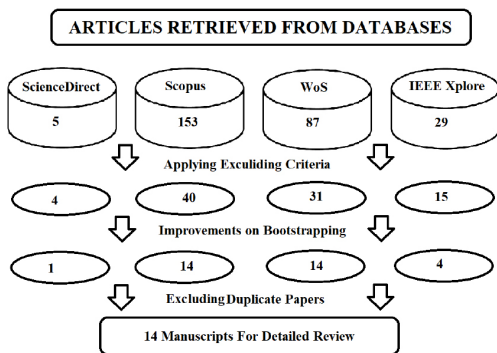


Figure 3

Selection of the articles

Table 3

Titles and year of publication of chosen articles

Article Title	Year	Ref.
Bootstrapping of FHE over the integers with large message space	2018	[43]
Fully homomorphic encryption over the integers for non-binary plaintexts without the sparse subset sum problem	2019	[5]
Privacy protection of IoT based on fully homomorphic encryption	2018	[60]
Matrix FHE and its application in optimizing bootstrapping	2018	[61]
Bootstrapping for HELib	2021	[29]
Efficient homomorphic integer polynomial evaluation based on GSW FHE	2018	[62]
Generalized Bootstrapping Technique Based on Block Equality Test Algorithm	2018	[67]
Analysis on matrix GSW-FHE and optimizing bootstrapping	2018	[66]
Efficient Fully Homomorphic Encryption with Large Plaintext Space	2018	[42]
Faster bootstrapping with multiple addends	2018	[68]
TFHE: fast fully homomorphic encryption over the torus	2020	[17]
Integer-Wise Functional Bootstrapping on TFHE: Applications in Secure Integer Arithmetics	2021	[49]
Improved homomorphic discrete Fourier transforms and fhe bootstrapping	2019	[30]
Near-optimal polynomial for modulus reduction using l2-norm for approximate homomorphic encryption	2020	[41]

In the next subsection we tabulated information for these mentioned 14 papers and give brief summary for each one of them.

3.2 Brief Summary of the Selected Studies

In Table 4, we give the essence of the papers. For interested readers, we use links to connect our table with brief summaries in the subsections.

Table 4
Improvements for the Bootstrapping Process

	Base Scheme	Hard Problem	Plaintext Space	Improvements For Bootstrapping
[43]	DGHV	AGCD SSSP	$\mathbb{Z}_Q$	Lagrange interpolation polynomial to reduce multiplication number (3.2.1)
[5]	DGHV	AGCD	$\mathbb{Z}_Q$	Efficient bootstrapping and reduced ciphertext size. (3.2.2)
[60]	BGV	RLWE	p^r	Better chosen parameter for efficient bootstrapping Fast bit-extraction (3.2.3)
[61]	BGV	LWE	M	Optimized bootstrapping key Better slot switching (3.2.4)
[29]	BGV	RLWE	p^r	Improved matrix multiplication algorithm Efficient noise management Smaller bootstrapping key Faster transformations Better digit extraction (3.2.5)
[62]	GSW	LWE	$\mathbb{Z}_Q^{r \times r}$	Horner's rule for faster and low noise growth in multiplication (3.2.6)
[67]	GSW	LWE	$\mathbb{Z}_Q^l$	Block test vector for efficiency and small bootstrapping key (3.2.7)
[66]	GSW	LWE	M	Improved bootstrapping key size Lowered complexity with hybrid homomorphic plaintext slot-wise switching (3.2.8)
[42]	FHEW	RLWE	$\mathbb{Z}_t$	Optimized most significant bit extraction (3.2.9)
[68]	TFHE	TLWE	$\mathbb{T}_N[X]$	Optimized paramaters and efficient multiplication (3.2.10)

Contd...

[17]	TFHE	TLWE	$\mathbb{T}_N[X]$	3 layer noise management in circuit bootstrapping Optimized CMux Improved accumulator Faster external product Efficient noise management Smaller bootstrapping key (3.2.11)
[49]	TFHE	TLWE	$\mathbb{T}$	Utilized test vector for blind rotation (3.2.12)
[30]	CKKS	RLWE	R	Cooley-Tukey matrix factorization for transformations Efficient multiplication algorithm Smaller bootstrapping key (3.2.13)
[41]	CKKS	RLWE	R	Better noise management via L2-norm minimization (3.2.14)

We exclude full disclosure of the related information in order to increase readability. We chose not to arrange all of the notations. In table, we employed $R = \mathbb{Z}[X]/\langle X^N + 1 \rangle$, $M \in \{0,1\}^{r \times r}$, Q and p be a prime, r, l, t, N are positive integers.

3.2.1 Bootstrapping of FHE over the Integers with Large Message Space

In [43], authors achieved to loosen parameter choosing criteria in [48]. To do that, they used Lagrange interpolation polynomial in the bootstrapping process. Specifically, the Lagrange interpolation polynomial is used to obtain a bit representation of an integer in mod Q . This new approach leads to a reduction of the number of multiplications to $\log(\theta \log \theta)$ from $O(\Theta \log^2 \theta)$ in [48]. So, the overall speed of the scheme is increased. Finally, the authors compare and conclude that their scheme performs better at large plaintext space.

3.2.2. Fully homomorphic encryption over the integers for non-binary plaintexts without the sparse subset sum problem

In [5], authors used reference [16] to build a FHE scheme over integers for non-binary plaintext in $\mathbb{Z}_Q$ for prime Q which is stated as Q-FHE-OI. Also, they removed the scheme's reliance on hardness of the SSSP (Sparse Subset Sum Problem). After expanding plaintext they redefined all procedures accordingly and achieved reduced ciphertext size complexity

from $O(\frac{\log Q}{Q^6})$ to $O(\frac{1}{\log Q})$. In the bootstrapping step they take advantage of newly defined Q -ary, Q -ary full adder, $(Q+1):2$ compressor and Q -ary round-to-nearest-integer functions and manage to lower time complexity from $O(\frac{\log^4 Q}{Q^7})$ to $O(\frac{1}{\log^2 Q})$, for $Q \geq 3$. They pointed out that their work is preferable for $Q \leq 11287$.

3.2.3 Privacy protection of IoT based on fully homomorphic encryption

In [60], authors considered the work of Halevi and Shoup [29] from 2015 and found ways to improve overall performance of the FHE scheme. The authors focused on the load capacity, which is related to fast bit extraction procedure. To improve the scheme, authors managed to find better parameters for the load capacity and get a slightly better fast bit extraction process. As a result, they manage to lower the bootstrapping time to 97 seconds from 172. Also, necessary hardware space is reduced from 3.0 GB to 2.3 GB for similar plaintext space and security level with HELib from 2015. In addition, they generalize the ciphertext space, so application areas of the scheme are extended.

3.2.4 Matrix FHE and its application in optimizing bootstrapping

Working on [32] which is FHE scheme on matrices, authors suggested optimization on bootstrapping. This new scheme, [61], supports homomorphic Hadamard product. In addition, for this scheme authors used test vector from [3] to achieve faster bootstrapping. Also, using this new definition authors obtained smaller ciphertext size from $(n+r) \times (n+r)$ $\lceil \log q \rceil$ to $r \times (n+r)$ and they obtained smaller bootstrapping key by a factor of $(\frac{n}{r+1}) \lceil \log q \rceil$ which helps them to decrease the bootstrapping time in comparison to paper [32].

3.2.5 Bootstrapping for HELib

This study [29] is HELib's, one of the main FHE schemes', update paper. The authors claim that their scheme reached to the point where it can be employed in real life applications. The re-encryption process takes less than 3 minutes for multiplicative depth 11 with security parameter 80. They state that scheme has been improved and analyzed in many important aspects especially things that affect the bootstrapping procedure are stated as follows. They improved the matrix multiplication algorithm and efficiency of the noise management. Also, they utilized tensor

decomposition for coefficient to slot and slot to coefficient steps. In addition, they employed Chen and Han's digit extraction technique to handle noise efficiently [13]. Also, they used 'thin bootstrapping' idea and added to their scheme which reduces computations in the bootstrapping process via using elements of the base structure instead of extension of base structure. As a result, the authors give numerous adjustments that help the scheme to be more robust, reliable and applicable.

3.2.6 Efficient homomorphic integer polynomial evaluation based on GSW FHE

This paper [62] is based on FHE scheme defined in [27] which is known as GSW. The authors reduced the integer multiplication noise using Horner's rule. Namely they used accumulator in $\mathbb{Z}_q$ and ciphertext in $\mathbb{F}_2^\ell$. In this setup, the process generates less noise per homomorphic integer multiplication with a ratio $O(\frac{(\log k)^w + 1}{nk^w})$ which increases the number of operations between bootstrappings.

3.2.7 Generalized Bootstrapping Technique Based on Block Equality Test Algorithm

In [67], authors used the test vector idea from [3] and improved it. A test vector is used to quicken the rounding step of bootstrapping procedure. They used a newly defined block test vector on SIMD scheme with a smaller bootstrapping key. They improved the multiplicative complexity of the paper [3] from $O(q)$ to $O(\log q)$ by employing a block homomorphic equality test algorithm. In addition, they claim that their work has better performance than the base test vector study with smaller bootstrapping key.

3.2.8 Analysis on matrix GSW-FHE and optimizing bootstrapping

In this GSW based FHE scheme [66], authors focused on circular security and using reject sample technique they optimized secret keys which improves the bootstrapping process. The core idea of the study is making homomorphic plaintext slot-wise switching by deterministic public encryption which outputs a reduced bootstrapping key. Also, with mentioned differences in the scheme they eliminated matrix multiplication in bootstrapping process and complexity of addition is increased according to compared study [32] $O(\log \lambda / \log(\log \lambda))$ to $O(\log^2 \lambda / \log(\log \lambda))$.

3.2.9 Efficient Fully Homomorphic Encryption with Large Plaintext Space

In [42], authors takes the open problem from the defining paper of FHEW and solved it by introducing adjusted and improved mechanisms for extended plaintext space. Apart from plaintext, they optimized the most significant bit extraction step and used it with a homomorphic accumulator. In the manuscript the authors calculated computational complexity as $O(N^3 \log^3 Q)$ which is the same as the base paper [23] but they claim that their scheme can perform better at large plaintext space.

3.2.10 Faster bootstrapping with multiple addends

Based on TFHE from 2016, the authors suggested a new scheme [68] and calculated a set of parameters which speed up the bootstrapping process. In addition, they defined enhanced homomorphic constant multiplication. The idea is to use a truth table and lower the needed homomorphic addition by half. As a result, they manage to bootstrap within 10 ms.

3.2.11 TFHE: fast fully homomorphic encryption over the torus

This work [17], is an update paper of TFHE, well known FHE scheme over the torus. We cannot fully summarize the work, instead we focus on bootstrapping procedure specifically. The authors give two different bootstrapping variants. First one is gate bootstrapping which means to bootstrap after every gate in the circuit. For this variant authors defined a faster external product and arranged the accumulator to improve the process. Also, their adjusted algorithm makes it possible to have a much smaller bootstrapping key. In addition, improvements in CMux leads to better noise management. Second variant is circuit bootstrapping which stands for bootstrapping once the noise level is reached to the threshold. For this approach, they defined 3 different noise management systems which have different structural properties. For comparison, they used the reference [23] and decreased running time of gate bootstrapping from 690 ms to 13 ms with smaller bootstrapping key with 16 MB in comparison to 1 GB.

3.2.12 Integer-Wise Functional Bootstrapping on TFHE: Applications in Secure Integer Arithmetics

In [49], authors suggested algorithms for homomorphic evaluation of integer arithmetics. Furthermore, they adjusted the coefficients of the test

vector which is used in the blind rotation step. But, they omitted the comparison or implementation generally. They are solely focused on integer arithmetics. As a negative aspect, the proposed scheme is impractical above 4-bit integers as an input.

3.2.13 Improved homomorphic discrete Fourier transforms and the bootstrapping

In the paper [30], authors achieved faster bootstrapping using Cooley-Tukey matrix factorization. They used this technique in the coefficients to slot and slot to coefficients steps in the bootstrapping process. The complexities of homomorphic constant multiplications and homomorphic rotations are reduced from $O(n)$ and $O(\sqrt{n})$ to $O(r \log_r n)$ and $O(\sqrt{r} \log_r n)$ where n is the message vector size and r is radix. After implementing on HEAAN library, they managed to lower the bootstrapping time of 26 hours to a mere 2 minutes on plaintext space $\mathbb{C}^3 2768$ and 8-bit precision where the size of bootstrapping key is reduced from 800 GB to 7 GB in the given settings.

3.2.14 Near-optimal polynomial for modulus reduction using l2-norm for approximate homomorphic encryption

In [41], authors used approximate homomorphic encryption as a base point [31] and modified it. In the bootstrapping process they employed L2-norm minimization instead of approximating the sine function at the modulus reduction step. As a result, this improvement leads to better noise management and reduces the bootstrapping frequency.

4. Discussion

In this section, we discuss the answers to our research questions stated at Table 1.

4.1 Which bootstrapping methodologies used in the FHE schemes ?

The bootstrapping methodologies in the articles can be observed according to underlying schemes. Our study contains 14 manuscripts with 6 different base schemes.

The first two papers are constructed on the DGHV. In [43], authors manage to lower the number of multiplication in the bootstrapping process, which leads to a better performing FHE scheme. The other paper based on DGHV is [5], the authors employed and extended the reference

[16] and managed to lower the complexity of the bootstrapping mechanism. In the reference SSSP reliance of the scheme is removed which is adopted in the paper.

The next base scheme is BGV. In [60], authors compare their results with HELib in 2015. The authors managed to improve the bit extraction step of the bootstrapping. In [61], authors defined a new scheme on matrices. Also, they constructed a smaller bootstrapping key, which made the process faster. The study [29] is an update paper of the HELib which contains earlier and latest improvements to the scheme. In the bootstrapping part, the authors improved the multiplication algorithm, obtained efficient noise management, used smaller bootstrapping key, presented faster transforms and gave a better digit extraction process. From the [60], we state the bootstrapping algorithm to present the process in the Table 5.

Table 5
Bootstrapping Algorithm of the BGV Scheme [50]

Input	$pk_2, D, \overline{< sk_{1i} >}, c$
Output	c'
1	$\overline{c_i} \leftarrow \text{HE.Enc}(pk_2)$
2	$c' \leftarrow \text{HE.Eval}(pk_2, D < \overline{< sk_{1i} >}, \overline{c_i} >)$

where (sk_i, pk_i) key-pairs of the scheme. In addition, c represents the ciphertext using pk , while sk_{1i} denotes the ciphertext of the i -th bit of sk_1 under the pk_2 . Moreover, HE.Enc and HE.Eval are the homomorphic encryption and homomorphic evaluation.

There are 3 manuscripts based on GSW. The first paper is [62] in which the researchers presented faster multiplication with low noise growth employing the Horner's rule. The second paper is [67]. The authors defined a block test vector and obtained efficiency with a small bootstrapping key. The third study is [66]. In it authors defined hybrid homomorphic plaintext slot-wise switching which lowers the complexity of the process. Also, they improved the bootstrapping key size. To present a bootstrapping algorithm in the Table 6, we refer to [62].

where bk is the set of keys for the bootstrapping. To be precise, $(C[i])_{i \in [0,l]}$ encrypts the $(\mu[i])_{i \in [0,l]} \in \mathbb{F}_2^l$. The G^{-1} is a randomized function which satisfies certain conditions.

The paper [42], is the only paper with FHEW base in our study. The researchers optimized the most significant bit extraction process, which is

Table 6
Bootstrapping Algorithm of the GSW Scheme [62]

Input	$bk, q = 2^l, C$
Output	$C[i]$
1	$i = 0$, while $i < l$, do
2	set $cvec$ as the $(N - 1 - i)$ -th column of C
3	$C[i] = \text{Bootstrap}(bk, cvec)$
4	$C = C - C[i] \bullet G^{-1}(M_{2^i})$
5	$i = i + 1$
6	end while

used as counterpart of the rounding step of the bootstrapping process. We restate the bootstrapping algorithm of this scheme in the Table 7. Actually, we merge two algorithms to form the following. The original algorithm handles the most significant digit extraction step separately.

Table 7
Bootstrapping Algorithm of the FHEW Scheme [42]

Input	s_i, B_r^j, c, κ
Output	$\mathbf{c}''$
1	$\text{ACC} \leftarrow b + \frac{q}{2^t}$
2	for $i = 1, \dots, n$, do
3	Compute $-a_i = \sum_j B_r^j \cdot a_{ij} \pmod{q}$ and get a_{ij}
4	end for
5	for $j = 0, \dots, d_r - 1$, do
6	$\text{ACC} \stackrel{\perp}{\leftarrow} K_{i, a_{ij}, j} = E(a_{ij} \cdot s_i B_r^j)$
7	end for
8	for $1 \geq k \geq t$, do
9	$\text{ACC}_{(k)} = \text{uMulticontant}(\text{ACC} \leftarrow k - 1)$
10	end for
11	for $1 \geq k \geq t$, do
12	$[(\mathbf{a}_{(k)})^t, \mathbf{b}_{(k)})^t] \triangleq [\vec{0}^t, (\mathbf{t}_{(k)})^t, \vec{0}^t, \dots, \vec{0}^t] \cdot \vec{\text{ACC}}_{(k)}$
13	end for
14	$c_{(k)} \triangleq (a_{(k)}, b_{(k)0}) \in \text{LWE}_{\frac{t}{z}}^{t/Q}(\text{msd}(\nu))$
15	$\mathbf{c} \triangleq \sum_{k=1}^{k=t} c_{(k)} \in \text{LWE}_{\frac{t}{z}}^{t/Q}(\text{msd}(\nu))$
16	$\mathbf{c}' \triangleq \text{KeySwitch}(\mathbf{c}, \kappa) \in \text{LWE}_s^{t/Q}(\text{msd}(\nu))$
17	$\mathbf{c}'' \triangleq \text{ModSwitch}(\mathbf{c}') \in \text{LWE}_s^{t/q}(\text{msd}(\nu))$

in here, B_r is the base of decomposition, s_i is the secret key and c is the ciphertext. The uMultcontant takes constant k with ciphertext C then outputs the ciphertext C_{ku} which is denoted by $C_{(k)}$. In addition, KeySwitch and ModSwitch are key switching and mod switching operations, respectively. Also, the msd term is stands for the most significant digit extraction operation.

The paper [68] is based upon the TFHE from 2017. The authors defined the enhanced homomorphic constant multiplication and achieves lower bootstrapping time. The update paper of the TFHE scheme is [17]. The authors give two different bootstrapping types, gate and circuit bootstrappings. For gate bootstrapping, researchers present optimized external product, improved accumulator and obtain smaller bootstrapping key. For the circuit bootstrapping, they defined 3 distinct levels of circuit in order to manage noise growth. The last TFHE based manuscript in our study is [49] in which authors generalized the coefficients of the test vector. The last bootstrapping algorithm we give is for the TFHE scheme. In the Table 8, we present the process from the study [68].

Table 8
Bootstrapping Algorithm of the TFHE Scheme [68]

Input	$(a, b), BK_s, msg = \mu_0, \mu_1$
Output	$LWE_{s,v} \begin{cases} \mu_0 & \phi_s(a, b) \in (-\frac{1}{4}, \frac{1}{4}] \\ \mu_1 & else \end{cases}$
1	$\bar{\mu} \triangleq \frac{\mu_0 + \mu_1}{2}, \bar{\mu}' \triangleq \frac{\mu_0 - \mu_1}{2}$
2	$\bar{b} \triangleq [2Nb], \bar{a}_i \triangleq [2Na_i], i \in [1, n]$
3	$testv = (1 + X + \dots + X_{N-1}) \times X^{-\frac{2N}{4}} \cdot \bar{\mu}' \in \mathbb{T}_N[X]$
4	$ACC \leftarrow (X^b \cdot (0, testv)) \in trivialTLWE_{a=0}(\pm \bar{\mu}' + \bar{\mu}' X \dots - \bar{\mu}' X^{N-1})$
5	for $i = 1 \rightarrow n/2$, do
6	$Keybundle_i = X^{-\overline{a2i-1}-\overline{a2i}} BK_{i,1} - X^{-\overline{a2i-1}} \cdot BK_{i,2} - X^{-\overline{a2i}} \cdot BK_{i,3} + BK_{i,4}$
7	$ACC \leftarrow Keybundle_i \square ACC$
8	end for
9	$u = (0, \bar{\mu}) + SampleExtract(ACC)$
10	$KeySwitch(u)$

where (a, b) is encryption of μ with secret key s and $\mu_0, \mu_1 \in \mathbb{T}$. BK is the bootstrapping key. The $Keybundle_i$ is a TGSW ciphertext obtained from the BK. Additionally, $SampleExtract$ is the operation of sample extraction from the ciphertext and $KeySwitch$ is the key switching operation.

The last base scheme is CKKS. The authors employed Cooley-Tukey matrix factorization in order to improve linear transformation steps in the bootstrapping process [30]. In [41], the approximate modulus reduction

step which calculates sine function is altered to L2-norm minimization process.

4.2 *What kind of updates are made to the bootstrapping process ?*

The bootstrapping process is the key mechanism of any FHE scheme. It is the most computationally intense part of the scheme. For the scope of our study, we investigate the papers in the Table 3 and their contribution to the process.

Some of the papers change the plaintext space and therefore adjusted the process using existing or new methods. We will mention those improvements in the following subsection.

An update to the bootstrapping can be made in the following four components.

- (1) Choosing parameters
- (2) Optimizing the bootstrapping key
- (3) Improving the inner parts of bootstrapping
- (4) Other

4.2.1 Choosing parameters

The FHE schemes rely on many different parameters. For security and optimization purposes, researchers try to obtain the best plaintext size, proper bootstrapping key.

The plaintext or message space is an important part of the scheme because the FHE schemes are planned to be used in the areas like IoT, database, banking, machine learning. All the areas need specific message spaces. For example, the machine learning process works well with approximate numbers. The schemes concerning IoT should be lightweight. Most of the papers present an optimized window for their parameters and compare with earlier results.

4.2.2 Optimizing the bootstrapping key

The bootstrapping key is a must in the process. The papers in this study, calculated and provided the smaller key in order to improve the overall performance. In some cases like [68], even though the bigger bootstrapping key is used, the overall process is faster. Note that, in [29] the bootstrapping key is chosen purposefully (as sparse as possible) to lower the complexity of the bootstrapping process.

Another aspect of the bootstrapping key is that it is an important part of the security of the scheme. The circular security of a scheme directly linked to the properties of the bootstrapping key, which we give details in the subsection 4.4.

4.2.3 Improving the inner parts of bootstrapping

The bootstrapping process and its work flow is differed from scheme to scheme. We name the inner parts of the bootstrapping process of the CKKS scheme and the HELib, noting the fact that different schemes have different inner parts and names.

The process for CKKS scheme has four steps, modulus raising, coefficients to slots, computing the mod function and slots to coefficients. The steps coefficients to slots and slots to coefficients are linear transformations, and many researchers try to increase the efficiency of the transformations. Especially for CKKS, the step, computing the mod function, is the core mechanism of the approximate scheme, which is why researchers try to find better performing methods.

The inner part of the HELib can be listed as modulus switching, multiplication with key, linear transformation, digit extraction, inverse linear transformation. Similar to the CKKS scheme, linear transformation steps are considered to be optimized. The digit extraction step is also one of the parts studied.

4.2.4 Other

There are numerous other improvements which effects performance of the schemes. To name one, we can mention the hoisting technique employed in the CKKS scheme. In [30], authors improved the efficiency by computing the common parts in the process last or first. Another improvement can be mentioned from the TFHE scheme, in [17], authors defined the gate and three level circuit bootstrappings which can be used in different situations.

4.3 *Which algebraic methods used in the bootstrapping process to increase efficiency?*

In the first prototype of homomorphic encryption scheme, the squashing technique is employed in order to simplify the bootstrapping process. Different approaches presented by the researchers, the generalized bootstrapping is one of them. The generalized bootstrapping, first defined in the [3]. The authors, provided two intertwined encryption schemes,

inner and outer encryption schemes. The idea is to perform decryption in the inner encryption scheme while encrypted and protected by the outer encryption scheme.

One of the important contributions to the bootstrapping process is done by the FHEW homomorphic encryption scheme. The authors introduced blind rotation technique which lowers the noise growth.

After mentioning some of the crucial methods that effect the bootstrapping process, we present some of the latest methods in the studies.

One of the methods is used in [30], authors employed the Cooley-Tukey matrix factorization in the linear transformation step of the bootstrapping process. By taking advantage of the well-studied Cooley-Tukey method, they manage to obtain a better performing bootstrapping process.

In [41], authors propose a new approach to the modulus reduction step of the bootstrapping process. The modulus reduction step is one of the major differences of the CKKS schemes. They use the L2-norm optimization method or in other words well-known least-squares method instead of employing sine function. With the proposed optimization, the necessity to bootstrap is lowered and noise management becomes better.

In [43], researchers leverage the Lagrange interpolation method. The method employed and multiplicative degree of the decryption circuit is lowered in comparison to the base scheme. As a result, using the Lagrange interpolation method, the authors manage to optimize the bootstrapping process.

Some other articles adjust the bootstrapping process to new plaintext space or employ different types of optimizations, which we try to address in the sections above.

4.4 What kind of key approaches used in the FHE schemes ?

In order to answer the question we approach it via their bootstrapping key.

The FHE schemes and their methodologies can be divided by their use of the bootstrapping keys. There are two different methods from this perspective.

- Single key approach to the FHE schemes
- Multi-key approach to the FHE schemes

The difference between single key and multi-key approaches is user numbers. In the single key approach, one user has the access to the plain text. It is not important who stores the data or makes operations on it. In multi-key approach, there are three types of schemes. First, the multi-key HE, in this setup, the number of keys can be changed according to the needs, but in the time of bootstrapping or decryption users should cooperate and obtain the evaluated result. Second, the threshold HE, in this type, the threshold limit is set before any computation and the keys are defined. If the number of users reached to the threshold and contribute to the process, then using information from the keys, the bootstrapping operation or the decryption can be made. Last, the hybrid multi-key HE, in which the dynamic nature of the multi-key HE and the efficiency of the threshold HE were converged, for more information see [2].

All of the articles in this study are based upon a single key approach to the FHE scheme, but we can investigate them further by the security assumptions. Some of the schemes have two different bootstrapping keys. To explain the situation, we need to recall the concept of the generalized bootstrapping process. The definition of the generalized bootstrapping includes two different encryption schemes, one inside of the other. To make the entire scheme able to operate homomorphic operations or lower the noise of the system, bootstrapping will occur safely inside of the outer encryption scheme which means inner encryption scheme is decrypted and reencrypted without giving away any important information.

For the key approaches of the FHE schemes, we can categorize them by the circular security assumption. The first definition of FHE scheme uses the circular security assumption [25]. Nowadays, most of the schemes do not rely on these criteria. The papers that use the assumption are [62, 66, 17]. The circular security assumption is a kind of KDM (Key Dependent Message) security. In the circular security assumption, the secret key bits can be encrypted without leakage of information. For more information about single and multi-key approaches and their KDM security, see [37].

5. Open Problems and Conclusion

In this section, we restate open problems which are clearly given in the papers. Also, we excluded 'bootstrapping can be improved' type statements.

- The authors think that combining 'best of both worlds' namely, HELib and TFHE libraries, is a very interesting open problem. Another

open problem is stated as; ‘Is there a more compact representation of elements of $\mathbb{Z}_{p^r}[X]$ that can be converted to DoubleCRT format in linear time?’.

- For GSW based scheme, authors employed Horner’s rule and stated that the study can be extended to multi-variate polynomials and Ring-GSW.
- For TFHE, authors stated three different open problems. First one is to find equivalence between RLWE and LWE. Second one is using DoubleCRT representation on TFHE settings. Third one is using multiplicative SIMD operations for TRGSW ciphertexts.
- New arithmetic operations such as multiplication, inversion and presentations are needed for parameter-based efficient implementations.
- New methods are required to find minimax polynomial and double angle formula-like approach.
- Block test vector can be used to improve efficiency of bootstrapping process.
- New techniques can be employed to achieve better performing DFTs (Discrete Fourier Transformation) which outcomes faster bootstrapping process.
- New techniques can be used to reduce memory requirements of FHE schemes.

For the conclusion, after giving fundamental information for some FHE schemes, we began the systematic literature review process. For it, we searched manuscripts about homomorphic encryption in the four major databases. After refinement, we obtained 14 articles that directly written for bootstrapping process. We tabulated core developments on the papers in regard to bootstrapping and summarized mentioned papers. Also, we answered our research questions in the Section 4. More specifically, we gave answers to questions like ‘Which bootstrapping methodologies used in the FHE schemes ?’, ‘What kind of updates are made to bootstrapping process of the FHE schemes ?’, ‘Which algebraic methods used in the bootstrapping process ?’ and ‘What kind of key approaches used in the FHE schemes ?’. Finally, we finished our study with open problems.

Acknowledgment

This study was partially supported by TUBITAK EEEAG under grant no.121F348.

References

- [1] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys (Csur)*, 51(4) : 1–35 (2018).
- [2] A. Aloufi, P. Hu, Y. Song, and K. Lauter. Computing blindfolded on data homomorphically encrypted under multiple keys: A survey. *ACM Computing Surveys (CSUR)*, 54(9):1–37 (2021).
- [3] J. Alperin-Sheriff and C. Peikert. Faster bootstrapping with polynomial error. In *Annual Cryptology Conference*, pages 297–314. Springer (2014).
- [4] S. M. Anggriane, S. M. Nasution, and F. Azmi. Advanced e-voting system using paillier homomorphic encryption algorithm. In 2016 International Conference on Informatics and Computing (ICIC), pages 338–342. IEEE (2016).
- [5] K. M. M. Aung, H. T. Lee, B. H. M. Tan, and H. Wang. Fully homomorphic encryption over the integers for non-binary plaintexts without the sparse subset sum problem. *Theoretical Computer Science*, 771:49–70 (2019).
- [6] P. B. Bomolo, E. M. Mukendi, and S. N. Badibanga. Trend of bootstrapping from 2009 to 2016. *International Journal of Advanced Computer Science and Applications*, 12(12) (2021).
- [7] D. Boneh, C. Gentry, S. Halevi, F. Wang, and D. J. Wu. Private database queries using somewhat homomorphic encryption. In International Conference on Applied Cryptography and Network Security, pages 102–118. Springer (2013).
- [8] D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-dnf formulas on ciphertexts. In TCC, volume 3378, pages 325–341. Springer (2005).
- [9] C. Bonte, I. Iliashenko, J. Park, H. V. Pereira, and N. P. Smart. Final: Faster fhe instantiated with ntru and lwe. *Cryptology ePrint Archive* (2022).
- [10] J. W. Bos, K. Lauter, J. Loftus, and M. Naehrig. Improved security for a ring-based fully homomorphic encryption scheme. In IMA In-

- ternational Conference on Cryptography and Coding, pages 45–64. Springer (2013).
- [11] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3) : 1–36 (2014).
 - [12] X. Che, T. Zhou, N. Li, H. Zhou, Z. Chen, and X. Yang. Modified multi-key fully homomorphic encryption based on ntru cryptosystem without key-switching. *Tsinghua Science and Technology*, 25(5) : 564–578 (2020).
 - [13] H. Chen and K. Han. Homomorphic lower digits removal and improved fhe bootstrapping. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 315–337. Springer (2018).
 - [14] J. H. Cheon, J.-S. Coron, J. Kim, M. S. Lee, T. Lepoint, M. Tibouchi, and A. Yun. Batch fully homomorphic encryption over the integers. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 315–335. Springer (2013).
 - [15] J. H. Cheon, A. Kim, M. Kim, and Y. Song. Homomorphic encryption for arithmetic of approximate numbers. In *International conference on the theory and application of cryptology and information security*, pages 409–437. Springer (2017).
 - [16] J. H. Cheon and D. Stehlé. Fully homomorphic encryption over the integers revisited. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 513–536. Springer, 2015.
 - [17] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. Tfhe: fast fully homomorphic encryption over the torus. *Journal of Cryptology*, 33(1):34–91 (2020).
 - [18] J.-S. Coron, T. Lepoint, and M. Tibouchi. Scale-invariant fully homomorphic encryption over the integers. In *International Workshop on Public Key Cryptography*, pages 311–328. Springer (2014).
 - [19] J.-S. Coron, A. Mandal, D. Naccache, and M. Tibouchi. Fully homomorphic encryption over the integers with shorter public keys. In *Annual Cryptology Conference*, pages 487–504. Springer (2011).
 - [20] J.-S. Coron, D. Naccache, and M. Tibouchi. Public key compression and modulus switching for fully homomorphic encryption over the integers. In *Annual International Conference on the Theory and Ap-*

- plications of Cryptographic Techniques, pages 446–464. Springer (2012).
- [21] R. Cramer, L. Ducas, C. Peikert, and O. Regev. Recovering short generators of principal ideals in cyclotomic rings. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 559–585. Springer (2016).
 - [22] M. v. Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In Annual international conference on the theory and applications of cryptographic techniques, pages 24–43. Springer (2010).
 - [23] L. Ducas and D. Micciancio. FHEW: bootstrapping homomorphic encryption in less than a second. In Advances in Cryptology–EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26–30, 2015, Proceedings, Part I 34, pages 617–640. Springer (2015).
 - [24] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472 (1985).
 - [25] C. Gentry. A fully homomorphic encryption scheme. Stanford university (2009).
 - [26] C. Gentry and S. Halevi. Implementing gentry’s fully-homomorphic encryption scheme. In Annual international conference on the theory and applications of cryptographic techniques, pages 129–148. Springer (2011).
 - [27] C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Annual Cryptology Conference, pages 75–92. Springer (2013).
 - [28] E. Giurgea, T. Hutu, and E. Simion. Some practical applications of fully homomorphic encryption. Cryptology ePrint Archive (2023).
 - [29] S. Halevi and V. Shoup. Bootstrapping for helib. *Journal of Cryptology*, 34(1) : 1–44 (2021).
 - [30] K. Han, M. Hhan, and J. H. Cheon. Improved homomorphic discrete fourier transforms and the bootstrapping. *IEEE Access*, 7 : 57361–57370 (2019).
 - [31] K. Han and D. Ki. Better bootstrapping for approximate homomorphic encryption. In Topics in Cryptology–CT-RSA 2020: The Cryptog-

- raphers' Track at the RSA Conference 2020, San Francisco, CA, USA, February 24–28, 2020, Proceedings, pages 364–390. Springer (2020).
- [32] R. Hiromasa, M. Abe, and T. Okamoto. Packing messages and optimizing bootstrapping in gsw-fhe. *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, 99(1) : 73–82 (2016).
 - [33] J. Hoffstein, J. Pipher, and J. H. Silverman. Ntru: A ring-based public key cryptosystem. In *International algorithmic number theory symposium*, pages 267–288. Springer (1998).
 - [34] N. Howgrave-Graham. Approximate integer common divisors. In *International Cryptography and Lattices Conference*, pages 51–66. Springer (2001).
 - [35] Y. Ishai and A. Paskin. Evaluating branching programs on encrypted data. In *Theory of Cryptography: 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21–24, 2007. Proceedings 4*, pages 575–594. Springer (2007).
 - [36] C. Kavitha and B. Harsoor. A survey on homomorphic encryption in cloud security. *International Journal of Engineering and Technology (UAE)*, 7(4) : 71–74 (2018).
 - [37] F. Kitagawa and T. Matsuda. Circular security is complete for kdm security. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 253–285. Springer (2020).
 - [38] K. Kluczniak. Ntru-nm: Secure fully homomorphic encryption from ntru with small modulus. *Cryptology ePrint Archive* (2022).
 - [39] B. S. Latibari, K. I. Gubbi, H. Homayoun, and A. Sasan. A survey on fhe acceleration. In *2023 IEEE 16th Dallas Circuits and Systems Conference (DCAS)*, pages 1–6. IEEE (2023).
 - [40] J.-W. Lee, H. Kang, Y. Lee, W. Choi, J. Eom, M. Deryabin, E. Lee, J. Lee, D. Yoo, Y.-S. Kim, et al. Privacy-preserving machine learning with fully homomorphic encryption for deep neural network. *IEEE Access*, 10:30039–30054 (2022).
 - [41] Y. Lee, J.-W. Lee, Y.-S. Kim, and J.-S. No. Near-optimal polynomial for modulus reduction using l2-norm for approximate homomorphic encryption. *IEEE Access*, 8:144321–144330 (2020).

- [42] N. Li, T. Zhou, X. Yang, Y. Han, and Y. Sun. Efficient fully homomorphic encryption with large plaintext space. *IETE Technical Review*, 35(sup1) : 85–96 (2018).
- [43] Z. Lian, Y. Hu, H. Chen, and B. Wang. Bootstrapping of fhe over the integers with large message space. *Security and Communication Networks*, 2018 (2018).
- [44] A. L’opez-Alt, E. Tromer, and V. Vaikuntanathan. On-the-fly multi-party computation on the cloud via multikey fully homomorphic encryption. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 1219–1234 (2012).
- [45] C. Marcolla, V. Sucasas, M. Manzano, R. Bassoli, F. H. Fitzek, and N. Aaraj. Survey on fully homomorphic encryption, theory, and applications. *Proceedings of the IEEE*, 110(10):1572–1609 (2022).
- [46] P. Martins, L. Sousa, and A. Mariano. A survey on fully homomorphic encryption: An engineering perspective. *ACM Computing Surveys (CSUR)*, 50(6) : 1–33 (2017).
- [47] M. Naehrig, K. Lauter, and V. Vaikuntanathan. Can homomorphic encryption be practical? In *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, pages 113–124 (2011).
- [48] K. Nuida and K. Kurosawa. (batch) fully homomorphic encryption over integers for nonbinary message spaces. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 537–555. Springer (2015).
- [49] H. Okada, S. Kiyomoto, and C. Cid. Integer-wise functional bootstrapping on tfhe: Applications in secure integer arithmetics. *Information*, 12(8) : 297 (2021).
- [50] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International conference on the theory and applications of cryptographic techniques*, pages 223–238. Springer (1999).
- [51] J. Park, D. S. Kim, and H. Lim. Privacy-preserving reinforcement learning using homomorphic encryption in cloud computing infrastructures. *IEEE Access*, 8 : 203564–203579 (2020).
- [52] H. V. L. Pereira. Bootstrapping fully homomorphic encryption over the integers in less than one second. In *IACR International Conference on Public-Key Cryptography*, pages 331–359. Springer (2021).

- [53] R. Podschwadt, D. Takabi, P. Hu, M. H. Rafiei, and Z. Cai. A survey of deep learning architectures for privacy-preserving machine learning with fully homomorphic encryption. *IEEE Access*, 10:117477–117500 (2022).
- [54] X. Qin, R. Huang, and H. Fan. An effective ntru-based fully homomorphic encryption scheme. *Mathematical Problems in Engineering*, 2021 (2021).
- [55] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6) : 1–40 (2009).
- [56] R. L. Rivest, L. Adleman, M. L. Dertouzos, et al. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11) : 169–180 (1978).
- [57] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2) : 120–126 (1978).
- [58] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee (1994).
- [59] A. Silverberg. Fully homomorphic encryption for mathematicians. *Women in Numbers 2: Research Directions in Number Theory*, 606 : 111 (2013).
- [60] W.-T. Song, B. Hu, and X.-F. Zhao. Privacy protection of iot based on fully homomorphic encryption. *Wireless Communications and Mobile Computing*, 2018 (2018).
- [61] B. Wang, X. Wang, R. Xue, and X. Huang. Matrix fhe and its application in optimizing bootstrapping. *The Computer Journal*, 61(12):1845–1861 (2018).
- [62] H. Wang and Q. Tang. Efficient homomorphic integer polynomial evaluation based on gsw fhe. *The Computer Journal*, 61(4) : 575–585 (2018).
- [63] Y. Yang, X. Huang, X. Liu, H. Cheng, J. Weng, X. Luo, and V. Chang. A comprehensive survey on secure outsourced computation and its applications. *IEEE Access*, 7:159426–159465 (2019).
- [64] Q. Zhang, L. T. Yang, Z. Chen, P. Li, and M. J. Deen. Privacy-preserving double-projection deep computation model with crowdsourcing on cloud for big data feature learning. *IEEE Internet of Things Journal*, 5(4) : 2896–2903 (2017).

- [65] F. Zhao, C. Li, and C. F. Liu. A cloud computing security solution based on fully homomorphic encryption. In 16th international conference on advanced communication technology, pages 485–488. IEEE (2014).
- [66] X. Zhao, H. Mao, S. Liu, W. Song, and B. Zhang. Analysis on matrix gsw-fhe and optimizing bootstrapping. *Security and Communication Networks*, 2018 (2018).
- [67] X. Zhao and A. Wang. Generalized bootstrapping technique based on block equality test algorithm. *Security and Communication Networks*, 2018 (2018).
- [68] T. Zhou, X. Yang, L. Liu, W. Zhang, and N. Li. Faster bootstrapping with multiple addends. *IEEE Access*, 6 : 49868–49876 (2018).

Received October, 2023