# A Survey of Homomorphic Encryption for Outsourced Big Data Computation

**Tan Soo Fun[1] and Azman Samsudin[1]**
[1] School of Computer Sciences, Universiti Sains Malysia
11800, Penang, Malaysia
[e-mail: soofuntan@gmail.com]
[e-mail: azman.samsudin@usm.my]
**Corresponding author: Azman Samsudin*

*Received April 26, 2015; revised April 17, 2016; revised June 8, 2016; accepted July 6, 2016; published August 31, 2016*

## Abstract

With traditional data storage solutions becoming too expensive and cumbersome to support Big Data processing, enterprises are now starting to outsource their data requirements to third parties, such as cloud service providers. However, this outsourced initiative introduces a number of security and privacy concerns. In this paper, homomorphic encryption is suggested as a mechanism to protect the confidentiality and privacy of outsourced data, while at the same time allowing third parties to perform computation on encrypted data. This paper also discusses the challenges of Big Data processing protection and highlights its differences from traditional data protection. Existing works on homomorphic encryption are technically reviewed and compared in terms of their encryption scheme, homomorphism classification, algorithm design, noise management, and security assumption. Finally, this paper discusses the current implementation, challenges, and future direction towards a practical homomorphic encryption scheme for securing outsourced Big Data computation.

***Keywords:*** Big Data Security, Homomorphic Encryption, Secure Outsourcing, Cloud Security.

This research was supported by a research grant from RUI, Universiti Sains Malaysia (USM) [1001/PKOMP/811334]. The author also thanks the anonymous reviewers of the paper for their careful reviews and valuable comments

**http://dx.doi.org/10.3837/tiis.2016.08.022** **ISSN : 1976-7277**

## 1. Introduction

The Big Data Era has arrived. Today, millions of enterprises are working on Big Data projects, from small start-ups to large multinational corporations. Currently, the amount of data being generated is doubling every 18 months, which represents grow that is faster than Moore's Law [1]. The study of the International Data Corporation (IDC) estimated that from 2013 to 2020, the digital universe will grow by a factor of 10 - from 4.4 trillion to 44 trillion gigabytes and the amount of data managed by enterprises will grow by 50 times. With traditional solutions becoming too expensive to support Big Data processing and shortage of Big Data analytics talent, enterprises are scrambling to outsource their data solution to third party service providers (i.e. cloud and mobile cloud computing) for cost saving and performance efficiency [2-3]. However, this outsourced initiative in turn introduces a number of security and privacy concerns. Enterprises are delegating direct access and control over their data to un-trusted third parties, who might be able to abuse their access to the data. For instance, Hadoop, which is the most popular Big Data processing platform, only employs the Kerberos authentication protocol for controlling Big Data access. Kerberos, however, does not encrypt an enterprise's data during data analysis and computation [4-6].

One of the recent alternatives for protecting outsourced Big Data is by installing tamper-resistant hardware in an un-trusted third party in order to prevent unauthorized access to the confidential data. Alternatively, enterprises can also encrypt their sensitive data with existing commercial Big Data protection tools (e.g. IBM InfoSphere Optim Data Masking, DataGuise, Cloudera Sentry, etc.) before sending it to third party service providers. However, an issue arises when there is a need to perform computations on confidential data (such as computing the frequency of words of the first ten billion documents, computing drug usage based on millions or tens of millions of patients' records, etc.), in which, both tamper-resistant hardware and existing encryption techniques are incompetent to support such operations. A trivial approach is that the enterprise can download the encrypted data (Big Data) and decrypt it before performing the analytical works, which is impractical and problematic.

Several recent advances in cryptography such as Private Information Retrieval (PIR), Searchable Encryption (SE), and Multi-Party Computation (MPC) schemes might be applicable to preserve the data privacy during data transformation. The PIR schemes [7-9] and SE schemes [10-14] enable enterprises to conduct searching of encrypted data, as well as retrieving private information securely, thereby, assuring the data security of un-trusted service providers. However, these schemes only support certain functionality, such as keyword search, ranking search, interval search, and subset search. The secure cloud storage schemes [15-18] that focus on executing various secure SQL statements over the outsourced databases can also be considered another alternative towards securing outsourced Big Data computation. Unfortunately, most of them [15-17] incur a high processing overhead during communication and decryption. Moreover, some of these schemes [10-17] leak the data access pattern, thus enabling third parties to learn from the search result [19-21]. On the other hand, the MPC protocol [22-25] allows a number of parties to jointly compute their desired algorithm based on the union of their data, without ever pooling or revealing their private data. The MPC protocol is well-designed for cooperative computation scenarios. A classic example of this protocol is when two or more competing corporates that jointly invest in a project that must satisfy both corporates' private and valuable constraints, are able to cooperate with each other to conduct computation tasks based on their desired private algorithm. Both TrustDB [26] and CipherBase [27] were developed to protect outsourced databases. However, they work on a co-design of hardware and software for specific customers with specific needs, thereby increasing the economic costs as compared to previous alternatives.

To overcome these problems, this paper suggests homomorphic encryption as the mechanism to protect confidential data, while allowing corresponding third parties to perform computation on encrypted data without having to decrypt it. The aim of homomorphic encryption is to ensure data privacy during communication, storage, or when in use by mechanisms similar to conventional cryptography, but with added capabilities of computing in relation to encrypted data [28-29]. Thus, homomorphic encryption makes Big Data computation outsourcing possible.

In this paper, we report a survey on homomorphic encryption for outsourced Big Data computation. Firstly, a conclusive study on Big Data as well as its definition and characteristics, and how it is different from traditional data protection, are presented in Section 2. Section 3 discusses the homomorphic encryption schemes - their definition, requirements, and classification. Besides that, existing works on homomorphic encryption are critically reviewed and compared in terms of encryption scheme, homomorphism classification, algorithm design, noise management scheme, and security assumption. The current implementation of homomorphic encryption, future research possibilities and challenges towards practical Fully Homomorphic Encryption (FHE) for securing outsourced Big Data computation are further discussed in Section 4. Finally, Section 5 concludes.

## 2. Big Data Characteristics and How it is Different from Traditional Data Security

The term “Big Data” can be traced back to the discussions by the academia and industry in the 80s on handling large groups of datasets [30-31]. Unfortunately, after all these years, Big Data is still in its early stages. Both academia and industry are still trying to comprehend its core nature and definition. For instance, the Oracle elaborates Big Data from the data type perspective [32]; meanwhile, the EMC/IDC research organization defines Big Data from the technologies and architectures perspective [33].

While academia and industry key players are struggling to define Big Data, in June 2013, the National Institute of Standards and Technology (NIST), took the leading role in the development of the Big Data technology roadmap and further defined Big Data as follows:

**NBD-PWG Definition**: *"Big Data refers to digital data volume, velocity and/or variety (veracity) that: i) Enable novel approaches to frontier questions previously inaccessible or impractical using current or conventional methods; and/or; ii) Exceed the capacity or capability of current or conventional methods and systems; iii) Differentiates by storing and analyzing population data and not sample sizes."* [34]

Generally, the definition of Big Data is associated with the three "V's". The first "V", Volume, refers to the scale of data (from terabytes to zettabytes). The McKinsey Global Institute reported that 90% of all data ever created, has been created in the past three years and data is growing exponentially [35]. Obviously, this tremendous data growth leads to the second "V", Velocity, which implies the importance of the data processing speed. The third "V", Variety, refers to different forms of data such as structured data, unstructured data, semi-structured data, quasi-structured data, etc. Recently, researches [1,31] have further expanded Big Data characteristics into five "V's", with an extra two "V's", namely: Veracity (also known as Verification or Variability) which refers to the uncertain nature of Big Data such as data consistency and trustworthiness, and Value which is an added-value that the collected data can bring to the intended process, activity, or predictive analysis. However, the nature of these "V's" remains within the measurement of data characteristics in itself. **Table 1** further summarizes the differences between Big Data and traditional data.

**Table 1.** The differences between Big Data and traditional data

| **Characteristic** | **Traditional Data** | **Big Data** |
|---|---|---|
| Volume | Bytes - megabytes | High Volume (Terabytes- Zettabytes) |
| Velocity | Moderate Speed | High Speed |
| Variety | Structured Data | Structure , Semi-Structured, Unstructured, "Quasi" Structured |
| Environments | Homogenous | Heterogeneous |
| Data Storage | Data are collected and stored in application/ service providers | High Scalability<br>Data are stored at data owner/data producer, only the aggregated resulted will stored at application/service providers. |
| Data Redundancy | High Redundancy | Low Redundancy |
| Security and Privacy Protection | Data At Rest<br>Data In Memory<br>Data In Transit | Data At Rest<br>Data In Memory<br>Data In Transit<br>Data in Transform |

Data protection is significantly different in the era of Big Data due to its associated characteristics. The high volume of Big Data implicates that the data is too expensive to be queried and moved into another resource for computation. Consequently, Big Data engineering is referred to as "moving the processing to the data, not the data to the processing" [36-37]. Previously, Big Data processing was limited only to a finite number of large enterprises and governments. Generally, their Big Data infrastructures were built in-house and typically isolated from public networks, therefore, security and privacy issues were not a concern. Today, more and more small-to medium-sized organizations are embracing Big Data to support their decision-making and, research and development (R&D) activities, and achieving competitive advantages in market share. According to the International Data Group (IDG) Big Data report [38], the average organization possessed an average of 164 terabytes of data during 2014. Over the next 12 to 18 months (2015-2016), that number was expected to increase by 76 percent, to 289 terabytes. Since enterprises' traditional data processing tools are becoming too expensive and limited to support Big Data processing, this rapid change has prompted enterprises to outsource their data storage, aggregation and analytical work to a third party [37].

These outsource computations, however, introduce a number of security and privacy concerns. This is because enterprises are delegating direct access and control over their data to an un-trusted third party, who might be able to abuse their access right to the data in order to infer, or more seriously sabotage, valuable information such as enterprises' intellectual property, trade secrets, financial information, etc. Thereby, the conventional cryptosystems that aimed to protect data-at-rest, data-in-memory, and data-in transit are insufficient to adapt directly to secure outsourced Big Data computations.

While SE, PIR, and MPC schemes focus on searching, retrieving, and joint-computing in relation to encrypted data respectively, the Homomorphic Encryption (HE) scheme – another scheme that is capable of protecting data during the transformation – is further suggested in this paper to be the appropriate mechanism for confronting security and privacy issues of outsourced Big Data computations. Therefore, with consideration of the Big Data characteristics of high volume, high velocity, and high variety, homomorphic encryption with speed efficiency is highly sought-after to protect data, and consequently making Big Data computing viable from the data security perspective.

## 3. Homomorphic Encryption (HE) Schemes and Related Works

HE schemes were originally known as privacy homomorphism, which was introduced by Rivest [39], shortly after the invention of the RSA cryptosystem. Even though this scheme was broken by Brickell and Yacobi [40] in 1988, the journey of finding an ideal homomorphic encryption was just about to begin.

### 3.1 Core Nature of Homomorphic Encryption Schemes

#### 3.1.1 Definition of a Homomorphic Encryption (HE) Scheme andHomomorphism Properties

Generally, a HE scheme allows computations to be performed on ciphertext without the need for the ciphertext to be decrypted. A HE scheme can be formally defined as follows:

**Definition 1**: A homomorphic encryption (HE) scheme is an encryption scheme, which has the following property for all $C_1$, $C_2 \in C$, $M_1$, $M_2 \in P$ and $K$, where $C_1 = Enc(M_1)$, $C_2 = Enc(M_2)$ and $C$, $P$ are groups:

$$Dec\ (C_1 \odot_C C_2) = M_1 \odot_P M_2 \qquad (1)$$

where $\odot_C$ and $\odot_P$ are the group operation in the ciphertext and plaintext space respectively.

If the plaintext space $P$ is an additive group, the homomorphic cryptosystem is known as additive homomorphism. Likewise, if the plaintext space is a multiplicative group, that homomorphic cryptosystem is considered as multiplicative homomorphism [29, 41].

#### 3.1.2 Symmetric Encryption Scheme vs. Asymmetric Encryption Scheme

Similar to conventional cryptography, the core nature of a HE scheme is to protect the confidentiality and privacy of data by encrypting the data either by using the same pair of secret keys (known as symmetric encryption scheme) or different pair of secret keys (known as asymmetric encryption scheme). Recent research on HEs are dominated by asymmetric encryption schemes [28, 39, 42-51, 52-57, 59-60, 62-78, 82, 84-86]. Only a few researchers [58, 61, 79-81, 83, 87, 88-91] have proposed HE schemes based on the symmetric algorithm. The main reason is the practical consideration for real-world deployments and the key management complexity of the symmetric crypto algorithm. Moreover, some of these symmetric homomorphic encryption schemes [58, 61, 79-80, 88-90] still suffer from security flaws in their algorithm design [94-98]. It has been argued that a stronger security proof is needed for these symmetric schemes [81, 83, 88-92].

### 3.1.3 Deterministic vs. Probabilistic

The core nature of the HE scheme design can be further categorized into deterministic or probabilistic depending on the probability properties of the encryption algorithm. In the deterministic scheme, for a fixed secret key, a given plaintext will always be encrypted into the same ciphertext. If the HE scheme is able to produce a different ciphertext, even with the same secret key and plaintext, then the encryption is categorized as probabilistic. In general, the probabilistic scheme enjoys a stronger security level as compared to deterministic scheme and most of the recent probabilistic HE schemes are recognized as semantic secure if an adversary is unable to obtain any useful information from the ciphertext in order to recover the corresponding plaintext [99]. From the literature, most of the recent works on HE schemes are semantically secure due to their probabilistic properties [42-51, 28, 52-85]. Only a few HE schemes [39] are considered to be deterministic HE schemes.

## 3.2 Homomorphic Encryption Scheme Classification and Related Works

Depending on the supported homomorphism properties and number of operations, HE schemes can be further classified into three categories, namely Partial Homomorphic Encryption (PHE), Somewhat Homomorphic Encryption (SWHE) and Fully Homomorphic Encryption (FHE). Their definitions and current progress are presented in the following subsections.

### 3.2.1 Partial Homomorphic Encryption (PHE)

PHE is defined as follows:

**Definition 2**: Partial Homomorphic Encryption (PHE) is either an additive homomorphism that supports only additive operations, or multiplicative homomorphism that supports only multiplicative operations.

Several conventional cryptosystems enjoy some sort of homomorphism. These schemes are categorized into PHE due to their computation limitation, that is, they are only able to conduct one type of computation operation on the encrypted data. For instance, the RSA scheme [39] supports only multiplicative homomorphism. Given RSA public key (*N*, *e*) and two plaintexts $M_1$ and $M_2$, the multiplication of two ciphertexts returns a computed multiplication in the encrypted domain as follows:

$$\begin{aligned} Enc(M_1) \times Enc(M_2) &\equiv (M_1^e \, mod) \times (M_2^e \, mod\, N) \\ &\equiv (M_1 \times M_2)^e \;\; mod\, N \\ &\equiv Enc(M_1 \times M_2) \end{aligned} \tag{2}$$

On the other hand, the Pai scheme [47] is limited to only additive homomorphism. Given the public key (*N*, *g*), a random number ($r_1$, $r_2$), and two plaintexts $M_1$ and $M_2$, the multiplication of two ciphertexts returns a computed addition in the encrypted domain as follows:

$$\begin{aligned} Enc(M_1) \times Enc(M_2) &\equiv [(g^{M1} . r_1^N) \times (g^{M2} . r_2^N)] \; mod\, N^2 \\ &\equiv [(g^{M1+M2})(r_1 \; r_2)^N] \; mod\, N \\ &\equiv Enc(M_1 + M_2) \end{aligned} \tag{3}$$

In the following, existing conventional cryptosystems that are considered as PHE schemes are listed in **Table 2**. A technical review in terms of their HE properties, algorithm design, message expansion, and the hardness of security assumptions (Integer Factorization problem (IF), Discrete Logarithm (DL) problem, e' the Root (eR) problem, Weak r'th Residue (wrR) problem, Subgroup Decision (SD) Problem, p-Subgroup (p-SP) problem, Quadratic Residuosity (QR) problem, Composite Residuosity (CR) problem, and Semantically Secure (SS)) is further presented. From **Table 2**, existing PHE schemes are constructed based on asymmetric encryption schemes and most of them [42-49, 51] enjoys non-deterministic properties, except for the RSA scheme [39] and MREA scheme [50]. Whereas majority of the existing PHE schemes [40-42, 44-51] are categorized as additive homomorphism, only the RSA scheme [39] and EGM scheme [43] support multiplicative homomorphism. In terms of their performance, the RSA scheme [39], EGM [43], and Pai [43] enjoy a higher efficiency.

Due to their performance efficiency, the majority of PHE schemes [39, 42-44, 47] are already being used in real-world applications. Most of these PHE schemes [39, 42-44, 47] are able to perform encryption and decryption in milliseconds. For example, the Pai scheme [47] which is widely applied in electronic voting protocols and biometric applications, takes approximately 2,313 milliseconds to encrypt and decrypt a 1024-bit data block. With their speed efficiency, these PHE schemes have been adapted for real-world Big Data application recently. CryptDB [100] that relies on a PHE scheme was commercialized in 2011. CryptDB executes an enterprise's queries regarding encrypted data in a MySQL database with a collection of adjustable query-based encryption schemes (e.g. the Pai encryption scheme [47] is used for supporting count query, Song *et al*.'s [10] encryption scheme is used to support keyword search query, etc.). Tu *et al*. proposed Monomi [101], by improving the performance of CryptDB and increasing the capability to handle a more complex query. Subsequently, several variants of CryptDB have been proposed recently, including MrCrypt [102], Crypsis [103], and Computing on Masked Data (CMD) [104]. MrCrypt [102] – focuses on MapReduce operation. Likewise, Crypsis [103] focuses on the high-level data flow language, Pig Latin. Similar to CryptDB, the additive homomorphism of both MrCrypt [102] and Crypsis [103] relies on the Pai scheme [47], whereas the EGM [43] scheme is used to support multiplicative homomorphism. On the other hand, CMD [104] further extends CryptDB into NoSQL databases.

However, to support different types of queries, every piece of data in CryptDB [100] and its variants [101-104] need to be encrypted under a different encryption schemes (e.g. Pai encryption scheme [47] is used for supporting count query, Song *et al*.'s [10] scheme or EGM [43] scheme are used to support keyword search query, etc.). These result in an increased data storage size (e.g. approximately 3.76 times in CryptDB). This overhead also increases the communication costs when outsourced data is transferred, back and forth, to third-party service providers.

Evidently, the limitation on the range of operations has limited PHE schemes from being accepted as solutions for Big Data outsourced computation [29, 41]. A few researchers have attempted to improve the versatility of existing PHE schemes. Examples of such effort include the MREA scheme [50] and CEG scheme [51]. However, the critical problem for both schemes is the approach of simply placing a plaintext as an exponent in order to support additive homomorphism. Firstly, both schemes do not support multiplicative homomorphism.

Table 2. A survey of Partial Homomorphic Encryption (PHE) schemes

| PHE Schemes | Encryption Scheme | | HE Properties | | | | | Security Assumption | | | | | | | | | Message Expansion |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Symmetric | Asymmetric | Probabilistic | Deterministic | Additive | Multiplicative | Algorithm | IF | eR | DL | QR | WrR | CR | SD | p-SP | SS | |
| RSA82 [39] | | √ | | √ | | √ | $(M_1^e)$ *mod N* × $(M_1^e)$ *mod N* $\equiv(M_1 \times M_2)^e$ *mod N* | √ | √ | | | | | | | | 1 |
| GM84 [42] | | √ | √ | | √ | | $(x^{M1} r_1^2) \times (x^{M2} r_2^2)$ *mod N* $\equiv[x^{M1+M2}(r_1\ r_2)^2]$ *mod N* | | | | √ | | | | | √ | *BitLength (N)* |
| EGM85 [43] | | √ | √ | | | √ | $(g^{r1}, M_1\ h^{r1}) \times (g^{r2}, M_2\ h^{r2})$ $\equiv[(g^{r1+r2}), (M_1\ M_2)\ h^{r1+r2}]$ | | | √ | | | | | | √ | 2 |
| Ben87 [44] | | √ | √ | | √ | | $(g^{M1}, r_1^b)$ *mod N* × $(g^{M1}, r_1^b)$ *mod N* $\equiv[(g^{M1+M2}), (r_1\ r_2)^b]$ *mod N* | | | | | √ | | | | √ | *BitLength (N)/b* |
| NS98 [45] | | √ | √ | | √ | | $(r_1^\sigma, g^{M1})$ *mod N* × $(r_2^\sigma, g^{M2})$ *mod N* $\equiv[(g^{M1+M2}), (r_1\ r_2)^\sigma]$ *mod N* | √ | | √ | | √ | | | | √ | ≥4 |
| OU98 [46] | | √ | √ | | √ | | $(g^{M1}h^{r1})$ *mod N* × $(g^{M2}h^{r2})$ *mod N* $\equiv(g^{M1+M2}\ h^{r1+r2})$*mod N* | √ | | | | | | | √ | √ | 3 |
| Pai99 [47] | | √ | √ | | √ | | $(g^{M1}\ r_1^N)$ *mod* $N^2 \times (g^{M2} r_2^N)]$ *mod* $N^2$ $\equiv[(g^{M1+M2})(r_1\ r_2)^N]$ *mod N* | | | | √ | | √ | | | √ | 2 |
| DJ01 [48] | | √ | √ | | √ | | $(g^{M1}\ r_1^{N^s})$ *mod* $N^{s+1} \times (g^{M2}\ r_2^{N^s})$ *mod* $N^{s+1}$ $\equiv[(g^{M1+M2})(r_1\ r_2)^{N^s}]$ *mod* $N^{s+1}$ | √ | | | | | √ | | | √ | (s+1)/s |
| BGN05 [49] | | √ | √ | | √ | | $(g^{M1}h^{r1})$ *mod N* × $(g^{M2}h^{r2})$ *mod N* $\equiv(g^{M1+M2}\ h^{r1+r2})$*mod N* | | | | | | | √ | | √ | *BitLength(N)/ BitLength(r)* |
| MREA12 [50] | | √ | | √ | √ | | $(g^{M1^s \bmod N}\ r^{M1})$ *mod* $N^2 \times (g^{M2^s \bmod N}\ r^{M2})$ *mod* $N^2$ $\equiv(g^{(M1+M2)^s \bmod N}\ r^{M1+M2})$*mod* $N^2$ | √ | | | | | √ | | | | ≥4 |
| CEG13 [51] | | √ | √ | | √ | | $(g^{r1}, h^{r1} g^{m1}) \times (g^{r2}, h^{r2} g^{m2})$ $\equiv[(g^{r1+r2}), (h^{r1+r2}\ g^{M1+M2}\ h^{r1+r2})]$ | | | √ | | | | | | √ | ≥4 |

Therefore, both schemes still have low versatility as original schemes. Secondly, the speed performance of both the MREA and CEG schemes is slower than their original schemes. Thereby, both the MREA and CEG schemes are still inadequate to be adapted directly for securing Big Data outsourced computation which requires high versatility and high speed.

### 3.2.2 Somewhat Homomorphic Encryption (SWHE)

A SWHE scheme is an encryption scheme that has some homomorphic properties but is not fully homomorphic, which is further defined as follows:

**Definition 3**: Somewhat Homomorphic Encryption (SWHE) is a homomorphic cryptosystem which has the ability to perform both additive and multiplicative homomorphism, however, with a limited number of operations. This limitation is governed by the cryptosystem's faintness to correctly decrypt resultant ciphertexts of homomorphic operations.

In general, each homomorphic encryption scheme outputs a ciphertext with a noise parameter, and the decryption works properly as long as the noise is less than the inherited security parameters. A SWHE scheme is able to support both additive and multiplicative homomorphism on encrypted data, however, with the compensation of increasing noise in the generated homomorphic ciphertext. Most of the HE schemes [28, 52-57, 59, 65-67, 101-106] proposed in the early twenty-first century are categorized into SWHE schemes due to the limited number of operations that can be executed in order to keep the noise parameters as small as possible. In other words, SWHE schemes can be used to support only a small subset of Big Data computations and certain real-world scenarios, such as protecting medical data [115], genomic and bioinformatics data [116-118], wireless sensor data [106], MySQL data [109], and Integer only data [108, 112, 113, 119]. Several researchers have adapted SWHE schemes to perform custom Big Data computations, including predictive analysis [115], regression analysis [114], statistical analysis [116, 119], and certain arithmetic operations [108].

### 3.2.3 Fully Homomorphic Encryption (FHE)

A FHE is similar to SWHE except that there is no increase in accumulated noise during computation. The general definition of a FHE scheme is further defined below:

**Definition 4**: Fully Homomorphic Encryption (FHE) supports both additive and multiplicative homomorphism which has the ability to perform an unlimited number of operations. Since addition and multiplication on any non-trivial ring constitute a Turing-complete set of gates, this scheme – if made efficient – allows one to employ any un-trusted computing resources without risk of revealing sensitive data [8, 28, 29].

The first FHE scheme [28] based on the ideal lattice approach was theoretically demonstrated by Gentry in 2009, after three decades of research exploration of homomorphic encryption. The technical review of existing FHE schemes in terms of their encryption scheme, homomorphism classification, algorithm design, noise management scheme, and security assumption are summarized in **Table 3** and **Table 4**.

From the literature, as summarized in **Table 3**, the majority of the existing FHE schemes [28, 52-57, 59, 65-67] are constructed from the SWHE with noise management techniques incorporated. Below is a list of noise management techniques used by FHE.

**Bootstrapping**. The first noise management technique introduced by Gentry in 2009 [28], in order to transform his SWHE scheme into a FHE scheme. Conceptually, bootstrapping is a technique extended from server-aided cryptography, where a ciphertext will be partially decrypted during the homomorphic computation in order to "refresh" and generate a new ciphertext with a low noise parameter. A bootstrapping technique involves a re-encryption algorithm and squashing algorithm. In the squashing algorithm, a "hint" of the secret key is encrypted as a portion of the public key. Whenever a ciphertext size or noise parameter grows too large during homomorphic computation, refreshing ciphertext with a re-encryption algorithm will be further applied. The ciphertext will be decrypted first and re-encrypted again using a "hint" of the secret key in order to produce a new encryption of the original plaintext, which is more compact and less noisy. However, the complex computations of squashing and re-encryption algorithms significantly result in slowing down the running speed of the FHE schemes. Existing FHE schemes that employ the bootstrapping technique to manage noise include [52-55, 57-58, 60, 62, 64, 70-72].

**Modulus Switching**. To solve the performance issues of bootstrapping, Brakerski and Vaikuntanathan [56] proposed a more lightweight technique to manage noise parameters during homomorphic computations. The modulus switching method does not fully refresh a ciphertext (as the re-encryption algorithm does), but successfully limits the noise growth in the ciphertext during homomorphic computations. Using a technique similar to the "dimension reduction" procedure [56], the evaluator is able to reduce the magnitude of the noise without knowing the secret key as in the bootstrapping technique. Instead, the evaluator only needs to know the ciphertext size bound in order to transform a ciphertext, *c* modulo *q* into a different ciphertext modulo *p* without sacrificing the correctness of the decryption procedure. As a result, this technique has a small ciphertext size as compared to the bootstrapping technique. Existing FHE schemes that use modulus switching to manage the noise include [59, 63, 65, 67-68, 73-75, 77-78, 84-85].

**Scale Invariant**. In 2012, Brakerski [66] introduced another interesting noise management technique, known as “Scale Invariant”. Placing a message space in the “lower bits” of the decryption equation requires the modulus switching technique to manage the noise. To tackle this problem, the message space can be placed in the “upper bits” of the decryption equation with the scale invariant technique, thereby controlling the noise growth in a more effective way. However, the limitation of the scale invariant technique is that a more complex rounding operation is required in multiplicative homomorphism [111]. Existing FHE schemes that apply the scale invariant technique include [69, 76].

**Flattening**. Recently, Gentry *et al*. [120] proposed the flattening technique which basically is a technique based on the modulus switching technique [56]. Generally, flattening is useful when the ciphertext is presented in matrix form and the encryption key is presented as a vector. The flattening technique uses a simple transformation [56] to modify the vectors without affecting the dot products, thus resulting in a better bound on the growth of the error. An existing FHE scheme that applies the flattening technique is [93].

**Noise-free FHE Scheme**. More recently, several researchers [61, 79-81, 82-83, 86-87] have proposed a Noise-free FHE scheme. These is a FHE schemes that does not use any noise management techniques as discussed previously. Instead of using the lattice, these Noise-free FHE schemes are constructed based on the classical number-theoretic concepts such as octonion algebra, commutative ring, and non-commutative ring. Evidently, the constructions of these Noise-free FHE schemes are dominated by the symmetric FHE [61, 79-81, 83, 86-87], although Nuida [82] proposed a Noise-free FHE scheme from the asymmetric perspective. However, some of these schemes [61, 79-81] are not secure. The Kipnis and Hibshoosh scheme [61] that is based on commutative rings is subject to known plaintext key-recovery attack [97-98]. Subsequently, both the Yagisawa scheme [79, 80] and Liu scheme [81] were proven to be not secure by Wang [96]. The Yagisawa scheme [86] that is based on the Discrete Logarithm problem cannot withstand quantum attack. On top of this, the Wang scheme [87] is only provable under the weak ciphertext-only security model. A detailed security analysis of these Noise-free FHE schemes is urgently sought after in order to establish a more reliable FHE scheme.

From the perspective of algorithm design, FHE schemes can be divided into three main categories: lattice-based, error correcting code-based and number theoretic-based. The detail of each category is further discussed in the following:

**Lattice**. Lattices are regular arrangements of points in Euclidean space. Lattice cryptography has been proved to have worst-case to average-case security with quantum reduction. As shown in **Table 3**, the majority of the existing FHE schemes [54-56, 59, 62-66, 68-69, 73-75, 78,84-85, 120] are constructed based on a lattice as found in Gentry's original work [28]. Under the lattice approach, the public key corresponds to the "bad" basis for a lattice, while the private key is the "good" basis of the same lattice. The lattice problems are conjectured to withstand quantum attacks. The main reason leading recent researchers to focus on lattices is because they can support both additive and multiplicative homomorphism. Besides that, the decryption circuit of a lattice is generally less complex as compared to former cryptosystems which involve exponentiation (e.g. RSA scheme [39], EGM scheme [43], etc.).

Depending on the inherited algorithm design and the hardness of the lattice problems (such as Polynomial Coset Problem (PCP), Ideal-Shortest Independent Vector Problem (I-SIVP), Sparse Subset-Sum Problem (SSSP), Bounded Distance Decoding Problem (BDDP), Learning With Error Problem (LWE), and Ring-Learning With Error (R-LWE)), these existing lattice-based FHE schemes can be further classified into three different classification approaches:

***Gentry***. To address the practicality issues of Gentry's FHE scheme [28], several empirical studies [52, 54-55, 60, 62, 70, 121] have been proposed over the last few years. In order to be practical, their research direction focused on improving the bootstrapping algorithm to speed up both encryption and decryption processes, while at the same time reducing the huge ciphertext and public key size. In 2010, Stehle and Sheinfeld [54] proposed a faster refreshing algorithm to improve the performance of the bootstrapping technique. Besides that, Gentry worked together with IBM researcher, Halevi [121], to optimize his original scheme by reducing the huge public key size and improve the performance of the primitives with a batching technique. However, these works are still far from being adequate to support practical applications. For encryption of one-bit plaintext, their method took more than a second to complete on a high-end 64-bit quad-core Intel Xeon E5450 64-bit processor, while re-encryption of primitives takes nearly half a minute for the lowest security setting (e.g. 380-bit size) [121]. In addition to the computation efficiency, the Gentry and Halevi scheme [121] requires a ciphertext of more than 780,000 bits for encryption of a single bit. This huge ciphertext size creates bottlenecks in bandwidths required to transfer the ciphertexts. Next, the bootstrapping technique without squashing the decryption circuit was proposed [55]. However, these schemes seem to pose rather inherent efficiency bottlenecks as they employ a bootstrapping technique from Gentry's original scheme in order to reduce the noise generated during homomorphic computations [74].

- ***Learning with Error (LWE) and R-LWE (R-LWE)***. In 2005, Regev defined the LWE problem as a generalization of "learning parity with noise" problem and proved that it enjoys similar worst-case hardness properties under a quantum reduction [126]. Subsequently, Brakerski and Vaikuntanathan [56] demonstrated the construction of FHE from this LWE problem. As compared to Gentry's approach that uses an algebraic notion of ideals in rings, the LWE assumption does not refer to ideals; indeed, the LWE problem is at least as hard as finding short vectors in any lattice. Several FHE schemes that are constructed based on the LWE problem, also known as "Regev Type FHE" schemes include [63, 66, 75, 84-85, 120]. More recently, other research groups [59, 62, 64, 65, 69, 72-74, 77-78, 93] have used a more efficient R-LWE in order to solve the quadratic overhead problem of LWE. The R-LWE is constructed based on ideal lattices as originally proposed by Lyubashevsky *et al.* [126], which is a special class of lattice that enjoys high efficiency compared to the general lattice groups. Both LWE and R-LWE based FHE schemes enjoy a special property, known as "additively key homomorphism", which allows the combination of encrypted data under different keys to produce an encryption (of the sum of the data) under the sum of the keys.
- ***NTRU***. NTRU was originally introduced by Hoffstein *et al*. in 1996, and firstly

applied by Stehle and Steinfeld [122] to improve the efficiency of FHE schemes. Generally, NTRU employs a R-LWE method to improve security assumptions based on worst-case hardness assumptions of the ideal lattices. This concept was further applied by Lopez *et al*. [65] in designing a multi-key FHE scheme. Unfortunately, these NTRU-based FHE schemes [54, 65, 69, 73, 78, 93, 122] still suffer from computation, bandwidth, and storage inefficiency due to their bit by bit encryption. Recent cryptanalysis [123-125] show that NTRU-based homomorphic encryption schemes are subject to key recovery attacks and are therefore not secure under the Chosen Ciphertext Attack (CCA-1) model.

**Table 3.** A classification of Fully Homomorphic Encryption (FHE) schemes in terms of their encryption scheme, algorithm design and noise management techniques

| FHE Scheme | Encryption scheme | | | | Algorithm Design | | | Noise Management Techniques | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Symmetric | Asymmetric | Probabilistic | Deterministic | Lattice | Number-Theoretic | Error Correcting Code | Bootstrapping | Modulus Switching | Scale Invariant | Flattening | Noise-Free |
| Gen10 [28] | | √ | √ | | √ | | | √ | | | | |
| SV10 [52] | | √ | √ | | | √ | | √ | | | | |
| DGHV10 [53] | | √ | √ | | | √ | | √ | | | | |
| SS10 [54] | | √ | √ | | √ | | | √ | | | | |
| GH11[55] | | √ | √ | | √ | | | √ | | | | |
| BV11 [56] | | √ | √ | | √ | | | | √ | | | |
| CMT11 [57] | | √ | √ | | | √ | | √ | | | | |
| BL11[58] | | √ | √ | | | | √ | √ | | | | |
| BGV12 [59] | | √ | √ | | √ | | | | √ | | | |
| Gu12[60] | | √ | √ | | | √ | | √ | | | | |
| KH12[61] | √ | | √ | | | √ | | | | | | √ |
| GHS12a [62] | | √ | √ | | √ | | | √ | | | | |
| GHS12b [63] | | √ | √ | | √ | | | | √ | | | |
| GHS12c [64] | | √ | √ | | √ | | | √ | | | | |
| LTV12 [65] | | √ | √ | | √ | | | | √ | | | |
| Bra12 [66] | | √ | √ | | √ | | | | | √ | | |
| CNT12 [67] | | √ | √ | | | √ | | | √ | | | |
| ZLX13 [68] | | √ | √ | | √ | | | | √ | | | |
| BLLN13 [69] | | √ | √ | | √ | | | | | √ | | |
| KLYC [70] | | √ | √ | | | √ | | √ | | | | |
| ZY13 [71] | | √ | √ | | | √ | | √ | | | | |
| CCKM13 [72] | | √ | √ | | | √ | | √ | | | | |
| GSW13[108] | | √ | √ | | √ | | | | | | √ | |
| DHS14 [73] | | √ | √ | | √ | | | | √ | | | |
| DSES [74] | | √ | √ | | √ | | | | √ | | | |
| CWS14 [75] | | √ | √ | | √ | | | | √ | | | |
| CLT14 [76] | | √ | √ | | | √ | | | | √ | | |
| ZW14 [77] | | √ | √ | | | √ | | | √ | | | |
| RC14 [78] | | √ | √ | | √ | | | | √ | | | |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Yag15a [79] | √ | | √ | | | √ | | | | | | √ |
| Yag15b [80] | √ | | √ | | | √ | | | | | | √ |
| Liu15 [81] | √ | | √ | | | √ | | | | | | √ |
| Niu15 [82] | | √ | √ | | | √ | | | | | | √ |
| LW15 [83] | √ | | √ | | | √ | | | | | | √ |
| CS15 [84] | | √ | √ | | √ | | | | √ | | | |
| DLLL16 [85] | | √ | √ | √ | | | | | √ | | | |
| Yag16 [84] | | √ | √ | | √ | | | | √ | | | |
| WANG16 [86] | √ | | √ | | | √ | | | | | | √ |
| DS16 [93] | | √ | √ | | | √ | | | | | √ | |

**Table 4.** A classification of Fully Homomorphic Encryption (FHE) schemes in terms of their Security Assumptions

| FHE Scheme | Hardness of Lattice Problems | | | | | | Hardness of Number-Theoretic Problem | | | | | | Harness of Decoding Random Linear code |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PCP | I-SIVP | SSSP | BDDP | LWE | R-LWE | A-GCD | DA-GCD | IF | DL | MQE | RTT | LPN |
| Gen10 [28] | | | √ | √ | | | | | | | | | |
| SV10 [52] | √ | | √ | | | | | | | | | | |
| DGHV10 [53] | | | | | | | √ | | | | | | |
| SS10 [54] | | | √ | √ | | | | | | | | | |
| GH11[55] | | √ | | | | | | | | √ | | | |
| BV11 [56] | | | √ | | √ | | | | | | | | |
| CMT11 [57] | | | | | | | √ | | | | | | |
| BL11[58] | | | | | | | | | | | | | √ |
| BGV12 [59] | | | | | | √ | | | | | | | |
| Gu12[60] | √ | | | | | | √ | | √ | | | | |
| KH12[61] | | | | | | | | | √ | | | | |
| GHS12a [62] | | | | | | √ | | | | | | | |
| GHS12b [63] | | | | | √ | | | | | | | | |
| GHS12c [64] | | | | | | √ | | | | | | | |
| LTV12 [65] | | | | | | √ | | | | | | | |
| Bra12 [66] | | | | | √ | | | | | | | | |
| CNT12 [67] | | | | | | | √ | | | | | | |
| ZLX13 [68] | | | | | | √ | | | | | | | |
| BLLN13 [69] | | | | | | √ | | | | | | | |
| KLYC [70] | | | √ | | | | | √ | | | | | |
| ZY13 [71] | | | | | | | √ | | | | | | |
| CCKM13 [72] | | | | | | √ | √ | | | | | | |
| GSW13[108] | | | | | √ | | | | | | | | |
| DHS14 [73] | | | | | | √ | | | | | | | |
| DSES [74] | | | | | | √ | | | | | | | |
| CWS14 [75] | | | | | √ | | | | | | | | |
| CLT14 [76] | | | | | | | √ | | | | | | |
| ZW14 [77] | | | | | | √ | | | | | | | |
| RC14 [78] | | | | | | √ | | | | | | | |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Yag15a [79] | | | | | | | | | | | √ | | |
| Yag15b [80] | | | | | | | | | | | √ | | |
| Liu15 [81] | | | | | | | √ | | | | | | |
| Niu15 [82] | | | | | | | | | | | | √ | |
| LW15 [83] | | | | | | | √ | | | | | | |
| CS15 [84] | | | | | √ | | | | | | | | |
| DLLL16 [85] | | | | | √ | | | | | | | | |
| Yag16 [84] | | | | | | | | | | √ | | | |
| WANG16 [86] | | | | | | | | | | | √ | | |
| DS16 [93] | | | | | | √ | | | | | | | |

**Error Correcting Codes**. While most of the existing FHE schemes [54-56, 59, 62-66, 68-69, 73-75, 78,84-85, 120] are directed toward the lattice, Bogdanov and Lee [58] proposed a new alternative to construct a FHE scheme from error correcting codes. In both code and lattice-based approaches, the message is encrypted by applying affine transformation and then noise is added to the message. In the lattice-based approach, the message is encrypted and hidden inside the noise, and therefore security is based on the inability to distinguish the different noise patterns, whereas, in the code-based approach, the message is hidden in the input of the affine transformation, in which the aim of the noise insertion is to prevent inversion. Unfortunately, Bogdanov and Lee's [58] scheme is subject to distinguisher-based attack [94]. Brakerski [95] on the other hand not only proved that Bogdanov and Lee's [58] scheme is not secure, but further argued that the entire approach of code-based homomorphic encryption is not secure.

**Number Theoretic**. The number theoretic approach has served as the fundamental basis in algorithm design for classical public key cryptography. This scheme enjoys some homomorphic properties and has been categorized as a PHE scheme as shown in **Table 2**. Over the last couple of years, the number-theoretic approach has begun to gain attention among researchers. They used the number theoretic approach to design a noise-free FHE scheme with the aim to root out the efficiency bottleneck of lattice-based FHE schemes. Existing FHE schemes that are constructed based on the number theoretic approach are further divided into four different categories as follows.

- ***DGHV***. A number of researchers have attempted to simplify Gentry's work into a number theoretic approach. In 2010, Dijk *et al*. [53] proposed another practical FHE alternative known as the DGHV (Dijk, Dijk, Gentry, Halevi, and Vaikuntanathan) scheme. The DGHV scheme is constructed based on elementary modular arithmetic rather than lattices as in Gentry's work [28] and its variants [52, 54-55, 60, 62, 70, 121]. The hardness of DGHV schemes is generally based on the Approximate Greatest Common Divisor problem (A-GCD) and Decision Approximate Greatest Common Divisor problem (DA-GCD). However, to achieve FHE, the DGHV scheme still uses the bootstrapping techniques from Gentry [28], thus limiting its performance. Recently, there have been works [67, 70, 72, 76] focusing on improving the efficiency of the DGHV scheme by reducing its public key size, as well as improving its bootstrapping technique. However, these schemes [53, 67, 70, 72, 76] are not considered to be Noise-free FHE schemes.

- ***Commutative and Non-Commutative Ring***. The idea of constructing FHE schemes from commutative rings was initially demonstrated by Kipnis and Hibshoosh [61] in

2012. It is a symmetric FHE scheme and its security is based on the hardness of Integer Factorization (IF) problem. Unfortunately, their scheme was proven to be not secure by Tsaban and Lishitz [98]. More recently, a few researchers [82, 83] investigated the construction of FHE schemes from the non-commutative ring perspective. Li and Wang [83] constructed a symmetric FHE from matrices over non-commutative rings. Meanwhile, Nuida [82] proposed asymmetric FHE by randomly applying Tietze transformations in order to conceal the concrete structures of the underlying non-communicative finite groups. Using the presentation of groups and Tietze transformation is still considered new in cryptology in which a comprehensive analysis of the reliability of their hardness assumption is further sought.

- ***Integer Vector***. In 2014, Zhou and Wornell [77] extended a LWE-based FHE scheme into integer vector with modulus switching, thus avoiding the complexity of classical bit-by-bit encryption found in FHE. However, this scheme is only able to support three fundamental operations on integer vectors: addition, linear transformation, and weighted inner products. In 2015, Liu [81] proposed another integer vector-based symmetric FHE scheme with the hardness of Approximate-Greatest Common Divisor (A-GCD). However, it was further broken by Wang in 2016 [87, 96].

- ***Octonion Algebra***. Instead of working on improving the noise management technique, several researchers [79-80, 86-87] are directed toward solving the root problem of FHE performance efficiency. They proposed a noise-free FHE based on octonion algebra over a finite ring, $\mathbb{Z}_{q,}$ with the security assumption that it is computationally infeasible to solve the Multivariate Quadratic Equation (MQE) systems. In 2015, Yagisawa [79-80] demonstrated how to construct a FHE scheme based on octonion algebra over finite rings. Both schemes [79, 80] are identical except for differences in the message encoding technique. Unfortunately, these schemes are proved to be not secure [87]. Subsequently, Yagisawa [86] proposed another octonion-based FHE scheme, with the security relying on the hardness of the classical Discrete Logarithm (DL) problem. More recently, Wang [87] proposed another noise-free FHE scheme which was based on octonion algebra. Wang's [87] scheme is only secure in weak ciphertext-only security mode, since the scheme cannot withstand adversaries who have access to sufficiently many linearly independent ciphertexts with known plaintexts and session randomness. Moreover, all of these schemes [78-80, 86-87] are constructed based on the symmetric FHE scheme. Whether it can be further extended into asymmetric FHE schemes is another interesting problem to be explored.

A comparison among PHE, SWHE and FHE schemes in terms of speed and versatility criteria is summarized in **Table 5**. It is undeniable that PHE [39, 42-51] perform faster than SWHE [105-119] and FHE [28, 52-93, 121]. However, these PHE schemes are only limited to additive homomorphism or multiplicative homomorphism. On the other hand, FHE schemes enjoy higher versatility as compared to SWHE and PHE schemes. Most existing FHE schemes use bootstrapping, dimension modulus reduction, scale-invariant or flattenning techniques. Complex computation has a negative impact on the speed performance of a FHE scheme. Whether these schemes can be further directly adapted to secure Big Data processing is still questionable.

**Table 5.** A Comparison of PHE, SWHE and FHE Scheme Based on Practical Homomorphic Scheme Criteria in Securing Outsourced Big Data Computation

| Homomorphic Encryption Scheme | Speed | Versatility |
|---|---|---|
| Partial Homomorphic Encryption (PHE) scheme | High | Low |
| Somewhat Homomorphic Encryption (SWHE) scheme | High | Medium |
| Fully Homomorphic Encryption (FHE) scheme | Low | High |

# 4. Towards a Practical Fully Homomorphic Encryption for Big Data Outsourced Computation

## 4.1 Recent Implementation of FHE schemes

While most FHE schemes are theoretically constructed, recently, there have been efforts to bring forward FHE for real-world application. The first implementation of a FHE scheme was demonstrated by Gentry and Halevi [121] in 2011. Unfortunately, the result was quite disappointing as their scheme took more than 900 seconds to add two 32-bit integers and more than 67,000 seconds to multiply them. To encrypt a single bit, the Gentry and Halevi FHE scheme [37] requires a ciphertext of more than 780,000 bits. The recent optimized implementation of FHE schemes is summarized into three general categories as follows.

**Parallelism Technique**. Several parallelism techniques such as packing ciphertext, batching processing, and Single-Instruction Multiple-Data (SIMD) operations are further applied to optimize the implementation of FHE schemes. The packing technique was firstly introduced by Smart and Vercauteren [52], and implemented by Gentry and Halevi [121]. Instead of working on bit-by-bit encryption, several bits of plaintext are “packed” into vector elements and encrypted into a single ciphertext with the application of the Chinese Remainder Theorem (CRT). This subsequently allows the batching technique to perform parallel homomorphic evaluation of a ciphertexts; the SIMD operations can be used for parallelizing the re-encryption algorithm, thus leading to substantial speeding-up and dramatically improving the required bandwidth and communication cost. For instance, the Coron *et al*. scheme [67] requires more than 74TB of encrypted data to be sent over the network for every 4MB of plaintext. However, with the parallelism technique [72], the communication requirement can be lowered to approximately 280GB [130]. Recent FHE schemes that use parallelism techniques to optimize their performance include [62-64, 69-74, 113, 127-132].

**Scheme Conversion Approach**. Conversion is used to reduce the bandwidth and storage size of ciphertext [63, 73-74, 130]. Generally, to speed up the re-encryption algorithm, plaintext will be encrypted with a symmetric encryption algorithm, which has a simple decryption circuit. If there is computation that needs to be carried out on the ciphertext, the decryption circuit of the targeted ciphertext will be evaluated homomorphically to re-encrypt the plaintext under the FHE scheme. Gentry *et al*. [63] were the first to demonstrate this technique by implementing their scheme [59] onto the Advanced Encryption Algorithm (AES) block ciphers. However, it took around 30 minutes to generate a single encryption key, and the corresponding encryption process of a1024-bit data block required over 36 hours of processing. In 2014, another group of researchers [73-74] repeated the work of Gentry *et al*.

[63], with NRTU-based homomorphic encryption scheme [65] and replaced AES with a more lightweight symmetric algorithm, Prince, and produced a scheme that took 79.5 minutes to encrypt a 1024-bit data block [74]. The performance speed of this scheme was better than the previous attempt [71]; however, the improved scheme is still impractical to be implemented for real-world application. A more promising result was demonstrated by Lepoint and Naehrig [130]. They implemented both a NTRU-based FHE scheme [69] and R-LWE-based FHE scheme [110]. The homomorphic evaluation was conducted with the lightweight block cipher, SIMON, which is extremely small, easy, and efficient to implement in hardware. The encryption algorithm method took approximately 2.7 minutes and 5 minutes to encrypt for the NTRU-based FHE scheme [69] and R-LWE-based FHE scheme [110] respectively. However, the homomorphic evaluation with SIMON-64/128 on a single core processor took approximately 3.4 hours and 4.58 hours for the NTRU-based FHE scheme [69] and R-LWE-based FHE scheme [110] respectively.

**Hardware Acceleration**. Instead of optimizing a FHE scheme from the software perspective as discussed above, another group of researchers [133-136] started work on accelerating FHE by implementing FHE in hardware. The first hardware implementation of a FHE scheme was demonstrated by Doroz *et al*. [133]. They designed a custom architecture for Gentry and Halevi's FHE scheme [121]. With several optimization techniques such as multi-million-bit multiplier based on Schnonhage Strassen multiplication and, spectral and precomputation strategy, the encryption, decryption and re-encryption algorithm per single bit took 18.1 milliseconds, 16.1 milliseconds, and 3.1 milliseconds respectively. Cao *et al*. [135] proposed the hardware implementation of two FHE schemes [57, 67] by using the FPGA technologies with the speed improvement of 44.73 and 54.2. FPGA hardware acceleration was further applied by Cousins *et al*. [136] and Poppelmann *et al*. [137] to improve the homomorphic computation for the NTRU-based FHE scheme and R-LWE-based scheme.

## 4.2 Future Directions of FHE in Securing Outsourced Big Data Computation

In this section, future research possibilities and the challenges of a practical FHE scheme for securing outsourced Big Data Computation are discussed.

**Hybrid or Non-Circuit Based.** Most of the recent FHE schemes were constructed according to circuit-based approach as inherited from Gentry's original work. The representation of each ciphertext as a single bit allows an arbitrary Boolean function to be computed simply with the binary addition and multiplication operation. However, the circuit-based approach has dramatically reduced the speed performance of the proposed FHE schemes and increased the communication bandwidth and storage size as compared to conventional cryptosystems. While it is difficult to construct a universal FHE scheme for general Big Data computations in practice, a hybrid or non-circuit based approach that is custom targeted to handle different scenarios of outsourced Big Data Computation such as Zhou and Wornell's scheme [77] is urgently sought after.

**Work with Access Control Mechanism**. A high volume of Big Data is being collected from diverse endpoints and accessed by different users. Whether FHE schemes can be further extended to enforce the access control on such data, so that authorized users are able to access sensitive data, while unauthorized users or malicious insiders are prohibited from obtaining this sensitive data, is another interesting topic to be explored. It includes working together with the decentralized key management approach, or taking advantages of the recent advancement of access control encryption schemes such as broadcast encryption scheme, Identity-Based Encryption (IBE) scheme, and Attribute-Based Encryption (ABE) scheme.

**Verifiable Homomorphic Encryption Scheme**. While FHE schemes can be used to protect the confidentiality of outsourced Big Data on un-trusted service providers, enterprises are concerned about the accuracy of computed results, and whether these un-trusted service providers are performing their actual work correctly or simply returning plausible results. To address this problem, homomorphic encryption can be used together with Verifiable Computation (VC) in order to guarantee the integrity of outsourced Big Data computations.

**Work with Artificial Intelligence Algorithm**. While the current works of improving the efficiency bottleneck of FHE are directed to algorithm design such as noise-free technique as well as, software and hardware implementation design, more recently, a few researchers from Microsoft [144] have demonstrated that machine learning algorithms can be applied to facilitate homomorphic computations. In CryptoNets [144], an enterprise's data are encrypted with R-LWE-based FHE scheme [69], and then an artificial feed-forward neural network is applied on the cloud service providers' side to speed up the homomorphic evaluation and make encrypted predictions. However, this neural network must be previously trained with a set of unencrypted data. The major limitation of this approach is the time needed to train the neural network efficiently. In the future, whether other artificial intelligence algorithms can be further applied to improve FHE performance is another interesting area to be explored.

**Alternative towards Gentry-based FHE Schemes**. Most FHE schemes are rooted in the Gentry's original blueprint. FHE schemes are generally constructed based on two phases – first, designing a SWHE scheme, and second, using the noise management techniques (e.g. bootstrapping, modulus switching, scale-invariant, and flattening) to transform SWHE schemes into FHE schemes. Therefore, how to achieve FHE directly without SWHE is still an open problem. The recent attempt to construct Noise-free FHE schemes based on the number theoretic approach (e.g. octonion algebra and non-commutative rings, etc.) seems to be another alternative to achieve practical FHE schemes. However, detailed security analysis is further required to prove their reliability.

**Secure FHE Scheme**. Recent cryptanalysis [94-98, 123-126, 138-141] has called for an urgent need to improve the security of the existing FHE schemes. The provable weak instance inherited from both LWE [139] and R-LWE [140-141] has resulted in some of the existing FHE schemes [54, 56, 59, 65, 73, 93, 122, 138] being vulnerable to attack. On top of this, almost all of the existing FHE schemes are only proved to be secure under the CCA-1 model, and whether FHE can further achieve CCA-2 is still questionable. In 2016, Gong *et al*. [142] attempted to construct a CCA-2 secure additive homomorphic encryption scheme, which was broken by Lee *et al*. [143] recently. Therefore, how to achieve a CCA-2 secure FHE scheme is still an open question to be solved.

## 5. Conclusion

Data protection in the era of Big Data is significantly different from that for traditional data due to the high volume, velocity, and variety of Big Data. In this paper, the homomorphic encryption scheme is suggested as the solution for securing outsourced Big Data computation.

Subsequently, the existing works on homomorphic encryption are reviewed and discussed in terms of their encryption approaches, algorithm design, noise management scheme, and the hardness of security assumptions. While lattice and number theoretical-based homomorphic encryption still suffer from unsatisfactory speed performance, storage, and bandwidth size, some researchers have already begun to look for hybrid approaches such as scheme conversion, parallelism and non-circuit based techniques. However, current advancements are still not adequate to secure Big Data processing. This paper also touches on the recent and upcoming research on homomorphic encryption, which focuses on addressing both the speed and versatility of Big Data processing. Whether these newer schemes can be further extended to overcome Big Data challenges is another important open issue for data scientists to delve into.

## References

[1] K. Krishnan, *Data Warehousing in the Age of Big Data*, CA: Morgan Kaufmann, 1st Ed., pp. 1-55, 2013. Article (CrossRef Link)

[2] A. V. Cardenas, P. K. Manadhata, and S. P. Rajan, "Big Data Analytics for Security," *IEEE Security and Privacy*, vol. 11, no. 6, pp. 74-76, Dec. 2013. Article (CrossRef Link)

[3] M. Cooper and P. Mell, "Big Data Technology and Implications for Security Research," in *Proc. of 19th ACM BADGERS Workshop*, North Carolina, pp. 15-16, 2012. Article (CrossRef Link)

[4] A. Becherer, "Hadoop Security Design Just Add Kerberos? Really?," *iSEC Partn.*, 2010. Article (CrossRef Link)

[5] O. O. Malley, K. Zhang, S. Radia, R. Marti, and C. Harrell, "Hadoop Security Design," *Apache Techn. Report*, 2009. Article (CrossRef Link)

[6] Zettaset, "The Big Data Security Gap : Protecting the Hadoop Cluster," 2013. Article (CrossRef Link)

[7] B. Chor, E. Kushilevitz, O. Goldreich and M. Sudan. "Private Information Retrieval," *JACM*, vol. 45, no. 6, pp. 965-981, 1998. Article (CrossRef Link)

[8] D. Boneh, E. Kushilevitz, R. Ostrovsk and W.E. Skeith III, "Public Key Encryption that Allows PIR Queries," *CRYPTO 2007*, *LNCS*, vol. 4622, pp. 50-67, 2007.Article (CrossRef Link)

[9] H. Avni, S. Dolev, N. Gilboa and X. Li, "SSSDB: Database with Private Information Search," *LNCS*, vol. 9511, pp.49-61, 2016. Article (CrossRef Link)

[10] D. Song, D. Wagner, and A. Perrig, "Practical Techniques for Searches on Encrypted Data," in *Proc. of IEEE Symposium on Secur. and Priv.*, pp. 44-55, 2000. Article (CrossRef Link)

[11] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records," in *Proc. of the ACM Workshop on Cloud Comput. Secur.*, pp.103-114, 2009. Article (CrossRef Link)

[12] Q. Liu, G. Wang and J. Wu, "Secure and Privacy Preserving Keyword Searching for Cloud Storage Services," *J. Netw. Comput. Appl.*, vol. 35, pp. 927-933, 2012. Article (CrossRef Link)

[13] K. Pasupuleti, S. Ramalingam, and R. Buyya, "An Efficient and Secure Privacy-preserving Approach for Outsourced Data of Resource Constrained Mobile Devices in Cloud Computing," *J. Netw. Comput. Appl.*, vol. 64, pp. 12–22, 2016. Article (CrossRef Link)

[14] S. Gajek, "Dynamic Symmetric Searchable Encryption from Constrained Functional Encryption," in *Proc. of CT-RSA 2016*, *LNCS*, vol. 9610, pp. 75-89, 2016. Article (CrossRef Link)

[15] R. Argawal, J. Kiernan, R. Srikant and Y. Xu, "Order Preserving Encryption for Numerical Data," in *Proc. of the ACM Special Interest Group on Management of Data(SIGMOD)*, 2004. Article (CrossRef Link)

[16] H. Hacigumus, B. Iyer, C. Li and S. Mehrotra, "Executing SQL Over Encrypted Data in the Database-Service-Provider Model," in *Proc. of the ACM Special Interest Group on Management of Data (SIGMOD)*, pp. 216-227, 2002. Article (CrossRef Link)

[17] B. Hore, S. Mehrotra and G. Tsudik, "A Privacy-preserving Index for Range Queries," in *Proc. of the ACM Symp. on Information, Compt. And Comm. Security (AISACCS)*, pp. 48-59, 2010.

Article (CrossRef Link)

[18] T. Xiang, X. Li, F. Chen, S. Guo and Y. Yang, "Processing Secure, Verifiable and Efficient SQL Over Outsourced Database," *Inform. Sci.*, vol. 348, pp. 163-178, 2016. Article (CrossRef Link)

[19] C. Liu, L. Zhu, M. Wan and Y.A. Tan, "Search Pattern Leakage in Searchable Encryption: Attacks and New Construction," *Inform. Sci.*, vol. 265, pp. 176-188, 2014. Article (CrossRef Link)

[20] M.S. Islam, M. Kuzu and M. Kantarcioglu, "Access Pattern Disclosure on Searchable Encryption: Ramification, Attack and Mitigation," Netw. *and Distr. Syst. Secur. Symp. (NDSS)*, 2012. Article (CrossRef Link)

[21] F.Han, J. Qin and J.Hu. "Secure Searches in the Cloud: A Survey," *Future Gener. Comput. Syst.* Accepted Manuscript. Article (CrossRef Link)

[22] A.Yao, "Protocol for Secure Computations," in *Proc. of 23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, pp. 160-164, 1982. Article (CrossRef Link)

[23] O. Goldreich, S. Micali and A. Wigderson. "How to Play any Mental Game – A Completeness Theorem for Protocols with Honest Majority," in *Proc. of 19th STOC*, pp. 218–229, 1987. Article (CrossRef Link)

[24] Y.Lindell and B.Pinkas, "Secure Multiparty Computation for Privacy-Preserving Data Mining," *The J. Priv. and Confi.*, vol. 1. no.1, pp. 59-98, 2009. Article (CrossRef Link)

[25] Damgard, A. Polychroniadou, V.Rao, "Adaptively Secure Multi-Party Computation from LWE (via Equivocal FHE)," *PKC-2016, LNCS*, vol. 9615, pp. 208-233, 2016. Article (CrossRef Link)

[26] S. Bajaj and R. Sion, "TrustedDB: A Trusted Hardware-based Database with Privacy and Data Confidentiality," *IEEE Trans. Knowl. Data. Eng.*, vol.26. no.3, pp. 752-765, 2014. Article (CrossRef Link)

[27] A. Arasu, K. Eguro, M. Joglekar, R. Kaushik, D. Kossmann and R. Ramanmurthy, "Transaction Processing on Confidential Data Using Cipherbase," in *Proc. of ICDE*, 2015. Article (CrossRef Link)

[28] C. Gentry, "A Fully Homomorphic Encryption Scheme," *Ph.D. Dissertation*, Dept. of Comp. Sci., Standford University, Stanford, CA , 2009. Article (CrossRef Link)

[29] J. Sen, "Homomorphic encryption - Theory and Application," *Theor. and Pract. of Crypt. and Netw. Secur. Protoc. and Techn.*, Croatia*: INTECH Publishers*, pp. 1–32, 2013. Article (CrossRef Link)

[30] M. Loukides, P. Warden, A. Watters, and A. Croll, *Big Data Now-Current Perspectives*, 1st ed. CA: O'Reilly Media, pp. 12-59, 2011. Article (CrossRef Link)

[31] J. Yan, "Big Data, Bigger Opportunities - Data.Gov's Roles: Promote, Lead, Contribute, and Collaborate in the era of Big Data," White Pap., 2013. Article (CrossRef Link)

[32] J. Dijcks, "Oracle: Big Data for the enterprise," White Pap., no. June, Oracle, 2013. Article (CrossRef Link)

[33] M. Gualtieri, "Evaluating Big Data Predictive Analytics Solutions," White Pap., Forrester, 2013. Article (CrossRef Link)

[34] NBP-PWG, "Big Data Definitions- v1," *NIST Big Data Public Working Group*, vol. 1, 2013. Article (CrossRef Link)

[35] J. Manyika, M. Chui, B. Brown, and J. Bughin, "Big data: The Next Frontier for Innovation, Competition, and Productivity," *McKinsey Global Institute*, 2011. Article (CrossRef Link)

[36] NBP-PWG, "NIST Big Data security and privacy requirements," NIST Big Data Public Working Group, vol. 1, 2013. Article (CrossRef Link)

[37] A. M. Cardenas, C. Yu, and Fuchs, "Expanded Top Ten Big Data Security and Privacy Challenges," CSA, 2013. Article (CrossRef Link)

[38] IDG Enterprise's Big Data Research, "IDG Enterprise Big Data Study 2014," *IDG*, 2014. Article (CrossRef Link)

[39] R. Rivest, "On Data Banks and Privacy Homomorphisms," *Found. of Secur. Comput.*, vol.4, no. 11, pp. 169-180, 1978. Article (CrossRef Link)

[40] E. Brickell and Y. Yacobi, "On Privacy Homomorphisms," *LNCS*, vol. 304, pp. 117-125, 1988. Article (CrossRef Link)

[41] J. M. Kukucka, "An Investigation of the Theory and Applications of Homomorphic," *Master Dissertation*, Dept. of Comp. Sci., Rensselaer Polytechnic Institute, Troy, NY, 2013. Article (CrossRef Link)

[42] S. Goldwasser and S. Micali, "Probabilistic Encryption," *J. Comput. Syst. Sci.*, vol. 28, no. 2, pp. 270-299, 1984. T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inf. Theory*, vol. 31, no. 4, pp. 469-472, Jul. 1985. Article (CrossRef Link)

[43] T. Elgamal, "A Public Key Cryptosystem and a Signature Scheme based on Discrete Logarithms," *IEEE Trans. Inf. Theory*, vol. 31, no. 4, pp. 469-472, Jul. 1985. Article (CrossRef Link)

[44] J. Benaloh, "Verifiable Secret-Ballot Elections," *Ph.D. Dissertation*, Yale University, 1987. Article (CrossRef Link)

[45] D. Naccache and J. Stern, "A New Public Key Cryptosystem based on Higher Residues," in *Proc. of 5th ACM Conf. on Comput. and Commun. Secur.*, San Francisco, pp. 59-66, 1998. Article (CrossRef Link)

[46] T. Okamoto and S. Uchiyama, "A New Public-key Cryptosystem as Secure as Factoring," *LNCS*, vol. 1403, pp. 308-318, 1998. Article (CrossRef Link)

[47] P. Paillier, "Public-key Cryptosystems based on Composite Degree Residuosity Classes," *LNCS*, vol. 1592, pp. 223-238, 1999. Article (CrossRef Link)

[48] I. Damgård and M. Jurik, "A Generalisation, a Simplication and Some Applications of Paillier's Probabilistic Public-key System," *LNCS*, vol. 1992, pp. 119-136, 2001. Article (CrossRef Link)

[49] D. Boneh, E. Goh, and K. Nissim, "Evaluating 2-DNF Formulas on Ciphertexts," *LNCS*, vol. 3378, pp. 325-341, 2005. Article (CrossRef Link)

[50] R. S. Dhakar, A. K. Gupta, and P. Sharma, "Modified RSA Encryption Algorithm (MREA)," in *Proc. of 2nd Int. Conf. Adv. Comput. Commun. Technol.*, Rohtak, pp. 426-429, 2012. Article (CrossRef Link)

[51] Y. Hu, "Improving the Efficiency of Homomorphic Encryption Schemes," *Ph.D. Dissertation*, Dept. Elect. and Comp. Eng., Worcester Polytechnic Institute, Worcester, MA, 2013. Article (CrossRef Link)

[52] N. Smart and F. Vercauteren, "Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes," *LNCS*, vol. 6056, pp. 325-341, 2010. Article (CrossRef Link)

[53] M. Van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully Homomorphic Encryption over the Integers," *LNCS*, vol. 6110, pp. 24-43, 2010. Article (CrossRef Link)

[54] D. Stehlé and R. Steinfeld, "Faster fully homomorphic encryption," *LNCS*, vol. 6477, pp. 377–394, 2010. Article (CrossRef Link)

[55] C. Gentry and S. Halevi, "Fully Homomorphic Encryption without Squashing Using Depth-3 Arithmetic Circuits," in *Proc. of IEEE 52nd Annu. Symp. Found. Comput. Sci.*, Palm Springs, pp. 107-109, 2011. Article (CrossRef Link)

[56] Z. Brakerski and V. Vaikuntanathan, "Efficient Fully Homomorphic Encryption from (Standard) LWE," in *Proc. of IEEE 52nd Annu. Symp. on Found. of Comp. Sci.*, CA, pp. 97–106, 2011. Article (CrossRef Link)

[57] J. Coron, A. Mandal, D. Naccache, and M. Tibouchi, "Fully Homomorphic Encryption over the Integers with Shorter Public Keys," *LNCS*, vol. 6841, pp. 487-504, 2011. Article (CrossRef Link)

[58] Andrej Bogdanov and Chin Ho Lee. "Homomorphic Encryption from Codes," Cryptology ePrint Archive, Report 2011/622, 2011. Article (CrossRef Link)

[59] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) Fully Homomorphic Encryption without Bootstrapping," in *Proc. of 3rd Innov. Theor. Comput. Sci. Conf.*, Cambridge, pp. 309-325, 2012. Article (CrossRef Link)

[60] C. Gu., "More Practical Fully Homomorphic Encryption," *Int. J. Cloud Comput. Serv. Sci.*, no. 4, pp. 1-17, Oct. 2012. Article (CrossRef Link)

[61] A. Kipnis and E. Hibshoosh, "Efficient Methods for Practical Fully Homomorphic Symmetric-key Encrypton, Randomization and Verification," *IACR Cryptol. ePrint Arch.*, pp. 1-20, 2012. Article (CrossRef Link)

[62] C. Gentry, S. Halevi, and N. Smart, "Better Bootstrapping in Fully Homomorphic Encryption,"

*LNCS*, vol. 7293, pp. 1-16, 2012. Article (CrossRef Link)

[63] C. Gentry, S. Halevi, and N. Smart, "Homomorphic evaluation of the AES circuit," *LNCS*, vol. 7417, pp. 850–867, 2012. Article (CrossRef Link)

[64] C. Gentry, S.Halevi, and N. Smart, "Fully Homomorphic Encryption with Polylog Overhead," *Advanced in Cryptology –EUROCRYPT 2012, LNCS*, vol. 7237, pp. 465-482, 2012. Article (CrossRef Link)

[65] A. López-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly Multiparty Computation on the Cloud via Multikey Fully Homomorphic Encryption," in *Proc. of 44th Symp. Theor. Comput., New York*, pp. 1219-1234, 2012. Article (CrossRef Link)

[66] Z. Brakerski, "Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP," *LNCS*, vol. 7417, pp. 868–886, 2012. Article (CrossRef Link)

[67] J. Coron, D. Naccache, and M. Tibouchi, "Public Key Compression and Modulus Switching for Fully Homomorphic Encryption over the Integers," *LNCS*, vol. 7237, pp. 446-464, 2012. Article (CrossRef Link)

[68] W. Zhang, S. Liu, and Y. Xiaoyuan, "RLWE-based Homomorphic Encryption and Private Information Retrieval," in *Proc. of 5th Int. Conf. on INCoS*, Xi An, pp. 535–540, 2013. Article (CrossRef Link)

[69] J.W. Bos, K. Lauter, J.Loftus, and M. Naehrig. "Improved Security for a Ring-based Fully Homomorphic Encryption Scheme," *LNCS*, vol. 8308, pp. 45-64, 2013. Article (CrossRef Link)

[70] J. Kim, M. Lee, A. Yun, and J. Cheon, "CRT-based Fully Homomorphic Encryption over the Integers," *IACR Cryptol. ePrint Arch*., pp. 1–18, 2013. Article (CrossRef Link)

[71] L. Zhang and Q. Yue, "A Fast Integer-based Batch Fully Homomorphic Encryption Scheme over Finite Field," *IACR Cryptol. ePrint Arch*., 2013. Article (CrossRef Link)

[72] J. Cheon, J. Coron, J. Kim, and M. Lee, "Batch Fully Homomorphic Encryption over the Integers," *LNCS*, vol. 7881, pp. 315-335, 2013. Article (CrossRef Link)

[73] Y. Doroz, Y. Hu, and B. Sunar, "Homomorphic AES Evaluation using NTRU," *IACR Cryptol. ePrint Arch*., pp. 1-16, 2014. Article (CrossRef Link)

[74] Y. Doröz, A. Shahverdi, T. Eisenbarth, and B. Sunar, "Toward Practical Homomorphic Evaluation of Block Ciphers using Prince," *LNCS*, vol. 8438, pp. 208-220, 2014. Article (CrossRef Link)

[75] Z. Chen, J. Wang and X. Song, "A Regev-Type Fully Homomorphic Encryption Scheme Using Modulus Switching," *The Scientific World Journal*, vol. 2014. Article (CrossRef Link)

[76] J. Coron, T. Lepoint, and M. Tibouchi, "Scale-Invariant Fully Homomorphic Encryption over the Integers," *LNCS*, vol. 8383, pp. 311-328, 2014. Article (CrossRef Link)

[77] H. Zhou and G. Wornell, "Efficient Homomorphic Encryption on Integer Vectors and Its Applications," in *Proc. of Info. Theor. And App. Workshop(ITA)*, pp. 1-9, 2014. Article (CrossRef Link)

[78] K. Rohloff and D. B. Cousins, "A Scalable Implementation of Fully Homomorphic Encryption Built on NTRU," *LNCS*, vol. 8438, pp. 221-234, 2014. Article (CrossRef Link)

[79] M. Yagisawa, "Fully homomorphic encryption without bootstrapping," *Technical report, IACR Cryptol. ePrint Arch*., Report 2015/474, 2015. Article (CrossRef Link)

[80] M. Yagisawa, "Fully homomorphic encryption on octonion ring," *Technical report, IACR Cryptol. ePrint Arch*., Report 2015/733, 2015. Article (CrossRef Link)

[81] D. Liu, "Practical fully homomorphic encryption without noise reduction," *Technical report, IACR Cryptol. ePrint Arch*., Report 2015/468, 2015. Article (CrossRef Link)

[82] K. Nuida, "Candidate Constructions of Fully Homomorphic Encryption on Finite Simple Groups without Ciphertext Noise," *IACR Cryptol. ePrint Arch*.,, Report 2014/97. Updated Nov.2015. Article (CrossRef Link)

[83] J.Li and L. Wang, "Noise-Free Symmetric Fully Homomorphic Encryption Based on Non-Commutative Rings," *IACR Cryptol. ePrint Arch*., Report 2015/614, 2015. Article (CrossRef Link)

[84] Z.Chen and X.Song. "A Multi-Bit Fully Homomorphic Encryption with Shorter Public Key from LWE," *IACR Cryptol. ePrint Arch*., Report 2015/1143, 2015. Article (CrossRef Link)

[85] Y.Ding, X. Li, H. Lu and X. Li, "A Novel Fully Homomorphic Encryption Based on LWE,"

Wuhan Uni. *J.of Natur. Scie.*, vol. 21, no.1, pp. 84-92, 2016. Article (CrossRef Link)

[86] M.Yagisawa, "Fully Homomorphic Public-key Encryption Based on Discrete Logarithm Problem," *IACR Cryptol. ePrint Arch.*, Report 2016/054, 2016. Article (CrossRef Link)

[87] Y. Wang, "Octonion Algebra and Noise-Free Fully Homomorphic Encryption (FHE) Schemes," *arXiv ePrint Archive Cornell University Library*, 2016. Article (CrossRef Link)

[88] I. Sharma, "Fully Homomorphic Encryption Scheme with Symmetric Keys," *Master Dissertation*, Dept. of Comp. Sci. and Eng., Rajasthan Technical University, Kota, Rajasthan 2013. Article (CrossRef Link)

[89] L. Xiao, O. Bastani, and I. Yen, "An Efficient Homomorphic Encryption Protocol for Multi-User Systems," *IACR Cryptol. ePrint Arch.*, pp. 1-19, 2012. Article (CrossRef Link)

[90] A. Chan, "Symmetric-Key Homomorphic Encryption for Encrypted Data Processing," in *Proc. of IEEE Comm. Societ.*, 2009. Article (CrossRef Link)

[91] P. Burtykam and O. Makarevich, "Symmetric Fully Homomorphic Encryption Using Decidable Matrix Equations," in Proc. of *7th Secur. of Info. and Netw*. 2014. Article (CrossRef Link)

[92] Y. Wang, K. She, Q. Luo, F. Yang and C. Zhao, "Symmetric Weak Ternary Quantum Homomorphic Encryption Schemes," *Mod. Phys. Lett. B, Accepted Manuscript*, 2016. Article (CrossRef Link)

[93] Y. Doroz and B. Sunar, "Flattening NTRU for Evaluation Key Free Homomorphic Encryption," *IACR Cryptol. ePrint Arch.*, Report 2016/315, 2016. Article (CrossRef Link)

[94] V.Gauthier, A. Otmani, and J-P, Tillich, "A distinguisher-based attack of a homomorphic encryption scheme relying on reed-solomon codes," *IACR Cryptol. ePrint Arch.*, Report 2012/168, 2012. Article (CrossRef Link)

[95] Z. Brakersi, "When Homomorphism Becomes a Liability," *Theory of Cryptography*, *LNCS*, vol. 7785, pp. 143-161, 2013. Article (CrossRef Link)

[96] Y. Wang, "Notes on Two Fully Homomorphic Encryption Schemes without Bootstrapping," *IACR Cryptol. ePrint Arch.*, Report 2015/519, 2015. Article (CrossRef Link)

[97] D. Vizar and S. Vaudenay, "Cryptanalysis of Chosen Symmetric Homomorphic Schemes," *Studia Scientiarum Mathematiarum Hungarica*, vol.52, no.2. 2015. Article (CrossRef Link)

[98] B. Tsaban and N.Lishitz, "Cryptanalysis of the MORE Symmetric Key Fully Homomorphic Encryption Scheme," *J. Math. Crypt*. Vol. 9, no. 2, pp.75–78, 2015. Article (CrossRef Link)

[99] C. Fontaine and F. Galand, "A Survey of Homomorphic Encryption for Nonspecialists," *EURASIP J. Inf. Secur.*, vol. 1, pp. 41–50, 2009. Article (CrossRef Link)

[100] R. Popa and C. Redfield, "CryptDB: Protecting Confidentiality with Encrypted Query Processing," in *Proc. of 23rd ACM SOSP-11*, pp. 85-100, 2011. Article (CrossRef Link)

[101] S. Tu, M. Kaashoek , S. Madden and N. Zeldovich, "Processing Analytical Queries over Encrypted Data," in *Proc. of VLDB Endowment*, pp. 289-300, 2013. Article (CrossRef Link)

[102] S. Tetali, M. Lesani, R.Majumar and T. Millstein. MrCrypt, "Static Analysis for Secure Cloud Computations," in *Proc. of 23rd ACM SIGPLAN-13*, pp.271-286, 2013. Article (CrossRef Link)

[103] J. Stephen, S.Savvides, R. Seidel and P. Eugster, "Practical Confidentiality Preserving Big Data Analysis," in *Proc. of 6th USENIX HotCloud-14*, pp. 10-16, 2014. Article (CrossRef Link)

[104] V. Gadepally, B. Hancock, B.Kaiser, J. Kepner, P.Michaleas, M. Varia and A. Yerukhimovich, "Computing on Masked Data to Improve the Security of Big Data," in *Proc. of IEEE Symposium HST-15*, pp.1-6, 2015. Article (CrossRef Link)

[105] M. Albrecht, J.-C. Faugere, P. Farshim, G. Herold, and L. Perret, "Polly Cracker, Revisited," *LNCS*, vol. 7293, pp. 17-33, 2011. Article (CrossRef Link)

[106] C. Castelluccia, "Efficient Aggregation of Encrypted Data in Wireless Sensor Networks," in *Proc. of 2nd Annu. Int. Conf. on MobiQuitous*, Coimbra, pp. 109-117, 2005. Article (CrossRef Link)

[107] L.V. Ly, "Polly Two-A Public Key Cryptosystem Based On Polly Cracker," *PhD Dissertation*, Faculty of Math., Ruhr-University Bochum, Bochum, 2002. Article (CrossRef Link)

[108] P. S. Pisa, M. Abdalla, and O.C. M. B. Duarte, "Somewhat Homomorphic Encryption Scheme for Arithmetic Operation," in *Proc. of Glob. Infor. Infrastr. and Netw. Symp (GIIS)*, 2012.

Article (CrossRef Link)

[109] D.Boneh, C.Gentry, S.Halevi, F.Wang and D.J. Wu, “Private Database Queries Using Somewhat Homomorphic Encryption,” *App. Crypt.and Netw. Secur.*, *LNCS*, vol. 7954, pp. 102-118, 2013. Article (CrossRef Link)

[110] J. Fan and F. Vercauteren, “Somewhat Practical Fully Homomorphic Encryption,” *IACR Cryptol. ePrint Arch.*, Report 2012/144, 2015. Article (CrossRef Link)

[111] A. Costache and N. P. Smart. “Which Ring Based Somewhat Homomorphic Encryption Scheme is Best?,” *CT-RSA 2016*, *LNCS*, vol. 9610, pp. 325-340. 2016. Article (CrossRef Link)

[112] M. Dijk, C. Gentry, S. Halevi and V. Vaikuntanathan, “Fully Homomorphic Encryption over the Integers,” *Adv. in Cryp- EUROCRYPT 2010*, *LNCS*, vol.6110, pp.24-43. 2010. Article (CrossRef Link)

[113] Y. Ramaiah and G. Kumari, “Efficient Public Key Homomorphic Encryption Over Integer Plaintexts,” in *Proc. of Infor. Sec.and Intell. Contr.*, pp. 123-128, 2012. Article (CrossRef Link)

[114] R. Hall, S. Fienberg and Y.Nardi, “Secure Multiple Linear Regression Based on Homomorphic Encryption,” *J. Official Statistics*, vol. 27, No. 4, pp. 669-691, 2011. Article (CrossRef Link)

[115] J. Bos, K.Lauter and M. Naehrig, “Private Predictive Analysis on Encrypted Medical Data,” *Microsoft Technical Reprt*, vol. 50, pp. 234-343, 2014. Article (CrossRef Link)

[116] K. Lauter, A. Lopez-Alt, M.Naehrig, “Private Computation on Encrypted Genomic Data,” *Progress in Crypt.- LATINCRYPT 2014*, *LNCS*, vol 8895, pp. 3-27, 2015. Article (CrossRef Link)

[117] N. Dowlin, R. G-Bachrach, K. Laine, “Manual for Using Homomorphic Encryption for BioInformatics,” *Microsoft Technical Reprt*, MSR-TR-2015-87, pp. 1-16, 2015. Article (CrossRef Link)

[118] M. Kim and K.Lauter, “Private Genome Analysis Through Homomorphic Encryption,” *BMC Medical Informatics and Decision Making*, pp. 1-15, 2015. Article (CrossRef Link)

[119] M. Yasuda, T. Shimoyama, J. Kogure, K. Yokoyama and T. Koshiba, “Secure Statistical Analysis Using R-LWE Based Homomorphic Encryption,” *Infor. Sec. and Priv*, *LNCS*, vol. 9144, pp. 471-487, 2015. Article (CrossRef Link)

[120] C. Gentry, A. Sahai, and B. Waters, “Homomorphic Encryption from Learning with Errors: Conceptually-simpler, Asymptotically-faster, Attribute-based,” *LNCS*, vol. 8042, pp. 72-92, 2013. Article (CrossRef Link)

[121] C. Gentry and S. Halevi, “Implementing Gentry’s Fully-Homomorphic Encryption Scheme,” *LNCS*, vol. 6632, pp. 129-148, 2011. Article (CrossRef Link)

[122] D. Stehle and R. Steinfeld, “Making NTRU as Secure as Worst-case Problems over Ideal Lattices,” *Adv. in Cryp- EUROCRYPT 2011*, *LNCS*, vol. 6632, pp. 27-47, 2011. Article (CrossRef Link)

[123] E. Morais and R. Dahab, “A Key Recovery Attack to the Scale-Invariant NTRU-based Somewhat Homomorphic Encryption Scheme,” *IACR Cryptol. ePrint Arch.*, Report 2014/898, 2014. Article (CrossRef Link)

[124] R.Dahab, S. Galbraith, and E. Morais, “Adaptive Key Recovery Attacks on NTRU-based Somewhat Homomorphic Encryption Schemes,” *Infor. Theor. Secur.*, *LNCS*, vol. 9063, pp.283-296, 2015. Article (CrossRef Link)

[125] M. Chenal and Q. Tang, “Key Recovery Attacks against NTRU-based Somewhat Homomorphic Encryption Schemes,” *Infor. Secur. LNCS*, vol. 9290, pp. 397-418, 2015. Article (CrossRef Link)

[126] V. Lyubashevsky, C. Peikert and O. Regev, “On Ideal Lattices and Learning with Errors over Rings,” *LNCS*, vol. 6110, pp. 1–23, 2010. Article (CrossRef Link)

[127] Z. Brakerski, C. Gentry and S. Halevi, “Packed Ciphertexts in LWE-based Homomorphic Encryption,” *PKC 2013*, *LNCS*, vol. 7778, pp. 1-13, 2013. Article (CrossRef Link)

[128] M. Yasuda, T. Shimoyama, J. Kogure, K. Yokoyama and T. Koshiba, “Practical Packing Method in Somewhat Homomorphic Encryption,” *LNCS*, vol. 8247, pp. 34- 50, 2014. Article (CrossRef Link)

[129] M. Yasuda, T. Shimoyama, J. Kogure, K. Yokoyama and T. Koshiba, “New Packing Method in

Somewhat Homomorphic Encryption and Its Applications," *J. Secur. and Comm. Netw.*, vol. 8, no. 13, pp. 2194- 2213, 2015. Article (CrossRef Link)

[130] T. Lepoint and M. Naehrig, "A Comparison of the Homomorphic Encryption Schemes FV and YASHE," *AFRICACRYPT 2014*, *LNCS*, vol. 8469, pp. 318-335, 2014. Article (CrossRef Link)

[131] N.P. Smart and F. Vercauteren, "Fully Homomorphic SIMD Operations," *Designs, Codes and Cryptography*, vol. 71, no. 1, pp. 57-81, 2014. Article (CrossRef Link)

[132] K. Nuida, K.Kurosawa, "(Batch) Fully Homomorphic Encryption over Integers for Non-Binary Message Spaces," *EUROCYPT 2015*, *LNCS*, vol. 9056, pp. 537-555, 2015. Article (CrossRef Link)

[133] Y. Doroz, E. Ozturk, and B. Sunar, "Accelerating Fully Homomorphic Encryption in Hardware," *IEEE Trans. On Comp*t., vol. 64, no. 6, pp. 1509 -1521, 2016. Article (CrossRef Link)

[134] Y. Doroz, E. Ozturk, E. Savas and B. Sunar, "Accelerating LTV Based Homomorphic Encryption in Reconfigurable Hardware," *CHES 2015*, *LNCS*, vol. 9293, pp. 185-204, 2015. Article (CrossRef Link)

[135] X. Cao, C. Moore, M. O'Neill, E.O'Sullivan and N. Hanley, "Accelerating Fully Homomorphic Encryption over the Integers with Super-size Hardware Multiplier and Modular Reduction," Cryptology ePrint Archive, Report 2013/616, 2013. Article (CrossRef Link)

[136] D. B. Cousins, J. Golusky, K. Rohloff and D. Sumorok, "An FPGA Co-Processor Implementation of Homomorphic Encryption," in *Proc. of IEEE High Perform. Extre. Comput.*, pp. 9-11, 2014. Article (CrossRef Link)

[137] T. Poppelmann, M. Naehrig, A. Putnam and A. Macias, "Accelerating Homomorphic Evaluation on Reconfigurable Hardware," *IACR Cryptol. ePrint Arch.*, Report 2015/631, 2015. Article (CrossRef Link)

[138] Y. Hu and F.Wang, "An Attack on a Fully Homomorphic Encryption Scheme," *IACR Cryptol. ePrint Arch.*, Report 2012/561, 2012. Article (CrossRef Link)

[139] K. Eisentrager, S. Hallgren and K. Lauter, "Weak Instance of PLWE," *SAC 2014*, *LNCS*, vol. 8781, pp. 183 -194, 2014. Article (CrossRef Link)

[140] Y. Elias, K. E. Lauter, E. Ozman and K. E. Stange, "Provably Weak Instances of Ring-LWE," *CRYPTO 2015*, *LNCS*, vol. 9215, pp. 63-92, 2015. Article (CrossRef Link)

[141] H. Chen, K. Lauter and K.E. Stange, "Attacks on Search-RLWE," *IACR Cryptol. ePrint Arch*, Report 2015/971, 2015. Article (CrossRef Link)

[142] L. Gong, S. Li, Q. Mao, D. Wang and J. Dou, "A Homomorphic Encryption Scheme with Adaptive Chosen Ciphertext Security But Without Random Oracle," *J. Theoret. Comput. Scien*. vol. 609, pp. 253- 261, 2016. Article (CrossRef Link)

[143] H. T. Lee, S. Ling and H. Wang, "Analysis of Gong et al.' s CCA2-Secure Homomorphic Encryption," *IACR Cryptol. ePrint Arch.*, Report 2016/019, 2016. Article (CrossRef Link)

[144] N. Downlin, R. G. Bachrach, K. Laine, K. Lauter, M. Naehrig and J. Wernsing, "CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy," *Microsoft Research Technical Report*, MSR-TR-2016-3, 2016. Article (CrossRef Link)

**Tan Soo Fun** is a Ph.D. candidate in School of Computer Sciences, Universiti Sains Malaysia(USM). She received her Bachelor of Information Technology(majoring in E-Commerce) and Master of Science (Computer Science) from Universiti Malaysia Sabah (UMS) in 2006 and 2009 respectively. Previously, she worked as a lecturer in School of Engineering and Information Technology at Universiti Malaysia Sabah. Her research interests include Cryptography, Information and Network security. She has published over 30 papers include book chapters, journals, technical reports and proceedings and received few research grants in the related fields. She is a member of Information Security Professional Malaysia (ISPA) and International Association of Computer Science and Information Technology (IACSIT). She is also a IBM Certified Academic Associate.

**Azman Samsudin** is a Professor at the School of Computer Science, Universiti Sains Malaysia (USM). He arned his B.Sc. in Computer Science from University of Rochester, New York, USA, in 1989. Later, he received his M.Sc. and Ph.D in Computer Science, in 1993 and 1998, respectively, both from the University of Denver, Colorado, USA. Recently, he serves as Deputy Dean of School of Computer Science, USM. His research interests include Crytography, Switching Networks and Parallel Computing. He has published more than 100 articles over a series of books, professional journals and conferences.